AD-A191 951

DTIC
S ELECTE
FEB 1 9 1988
∞ D

MODELING AND EVALUATION OF EXPERT SYSTEMS

IN DECISIONMAKING ORGANIZATIONS

Didier Marie-Joseph Perdu

Laboratory for Information and Decision Systems

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

88 2 4 03 2

DECEMBER 1987

LIDS-TH-1726

MODELING AND EVALUATION OF EXPERT SYSTEMS
IN DECISIONMAKING ORGANIZATIONS

by

DIDIER MARIE-JOSEPH PERDU

This report is based on the unaltered thesis of Didier M.-J. Perdu, submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Technology and Policy at the Massachusetts Institute of Technology in December 1987. The research was conducted at the MIT Laboratory for Information and Decision Systems with support provided by the Basic Research Group of the Technical Panel on C^3 of the Joint Directors of Laboratories, through the Office of Naval Research, under Contract No. N00014-85-K-0782.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

MODELING AND EVALUATION OF EXPERT SYSTEMS
IN DECISIONMAKING ORGANIZATIONS

by

DIDIER MARIE-JOSEPH PERDU

Ingénieur de l'Ecole Supérieure d'Electricité
(July 1986)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN TECHNOLOGY AND POLICY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
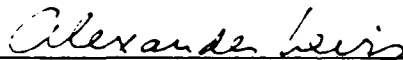
December 1987

Signature of Author_____
Department of Electrical Engineering and Computer Science
December 18, 1987

Certified by_____
Dr. Alexander H. Levis
Thesis Supervisor

Accepted by _____
Professor Richard de Neufville, Chairman
Technology and Policy Program

Accepted by _____
Professor Arthur C. Smith
Chairman, Departmental Graduate Committee
Electrical Engineering and Computer Science Department

MODELING AND EVALUATION OF EXPERT SYSTEMS
IN DECISIONMAKING ORGANIZATIONS

by

DIDIER M. PERDU

Submitted to the Department of Electrical Engineering and Computer Science
on December 18, 1987
in partial fulfillment of the requirements for the degree of
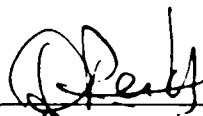Master of Science in Technology and Policy.

ABSTRACT

The introduction of expert systems as decision aids in decisionmaking organizations will modify their performance. First, a model of symbolic computation with fuzzy logic, using Predicate Transition Nets, is presented. The basic operators AND, OR, and NOT are then used to model the most common kind of expert systems : the consultant expert system in which production rules are used for knowledge representation. This model allows to simulate the dynamical behavior of the expert system in its search for a solution and to evaluate its response time for a given input. This response time depends on the number of rules scanned by the system and on the number of interactions with the user. An Air Defense Command and Control application, involving a hierarchical organization, where the expert system is used as an aid in the fusion of inconsistent information, is then developed. A strategy involving the use of the expert system is compared to two other strategies expected to be used by a decisionmaker facing this problem. Measures of performance (workload, timeliness, and accuracy) are evaluated for each of these strategies. The results show that the use of the expert system improves significantly the accuracy of the organization, but requires more time and increases the workload of the decisionmaker using it.

Thesis Supervisor : Dr. Alexander H. Levis
Title : Senior Research Scientist

4

ACKNOWLEDGEMENTS

I wish to express my gratitude to :

Alexander Levis for his remarkable supervision in the realization of this thesis. Working with him has really been a fruitful experience both professionally and personally.

Lisa Babine, for her kindness and her help in the final "polishing" of this thesis.

Jean-Marc, Jean-Louis, Anne-Claire and Louise for their comradeship and advice.

All my family and especially my parents and grandparents for their devoted support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Decisionmaking processes require the analysis of complex situations and the planning, initiation and control of subsequent responses. These activities are done within some constraints such as time and accuracy and so that an acceptable level of effectiveness be reached. The amount of information handled by decisionmakers is often very large and, in order to maintain performance above a certain level, decisionmaking organizations use decision support systems to help them accomplish their mission.

Artificial Intelligence offers many tools to make the decisionmakers' task easier. In a military context, the area covered by these tools range from the identification of targets to the planning of actions against a certain threat given a particular distribution of the available resources. Expert Systems are one of these tools. Their ability to handle symbolic concepts and their deductive capability make them very useful to a decisionmaker.

Expert systems are studied in this thesis to assess their usefulness in aiding the fusion of information coming from different sources and which is not always consistent. Expert Systems have the ability to combine pieces of evidence of different kinds and to make inferences using knowledge from their own higher levels of information. They can even deal with uncertain data using different methods. Their role in the information fusion process can be therefore critical. The main goal of this thesis is to investigate how the use of expert systems as decision aids in the information fusion stage modifies the performance of an organization.

1.2 BACKGROUND

The decisionmaker using an expert system will be studied with the analytical framework developed by Levis (1984). The analysis of the workload in decisionmaking organizations has been developed within the information theoretic framework introduced by Shannon (Shannon and Weaver, 1949). The analysis of the architecture is done with the Petri Net

formalism (Peterson, 1980; Reisig, 1985) which allows the modeling of asynchronous and concurrent processes.

Boettcher and Levis (1982-a) have developed a model of a single decisionmaker (DM). The DM has several alternatives to choose from in performing the assigned task. He is memoryless and has no capability to learn while executing a task, and is in an hostile environment where the tempo of operations is fast. These conditions imply that the decisionmaker acts subject to some constraints (time, accuracy) which have an important effect on the steady-state functioning of the organization. The value of the outcome of the actions performed in this hostile environment is expressed in terms of a cost for the organization. This model has been extended to the general case of interactions among several decisionmakers of an organization (Boettcher and Levis, 1982-b). The assumption that the decisionmaker is memoryless was relaxed by Hall and Levis (1984). Some studies have addressed the integration of preprocessors (Chyen and Levis, 1985) and information storage in decisionmaking organizations (Bejjani and Levis, 1985). This thesis continues the study of the effects of integrating decision aids in decisionmaking organizations.

1.3. THE THESIS IN OUTLINE

To show to what extent the performance of an organization is modified by the use of an expert system, two directions are pursued. The first one is the modelling of an expert system using fuzzy logic to deal with uncertainty. This model is based on the Predicate Transition Net representation and allows, through simulation capabilities, to make time-related measures (response time). The second direction is an approach to the problem of fusion of inconsistent or contradictory information through the use of expert systems. The strategy involving the use of expert systems is compared to other strategies expected to be used in the data fusion process. An application has been developed to illustrate measures of performance (accuracy, workload, and timeliness) and to give a precise idea of how this problem can be approached and solved using different strategies.

In chapter 2, the different tools used in this thesis for modeling and measuring are described. Information Theory and Petri Net Theory are described briefly. Attention is focused on Predicate Transition Nets which are used later in the thesis for the modelling of expert systems. The model of the interacting decisionmaker is finally reviewed along with the different measures of performance. The main concepts of knowledge based expert systems

14

are reviewed in chapter 3. This chapter describes also how uncertainty can be handled, especially with fuzzy logic and fuzzy set theory. The model of expert system with fuzzy logic is given in chapter 4 after the introduction of Predicate Transition Net models of the fuzzy logic operators that realize the operations AND, OR and NOT. Finally, the method used to evaluate time-related measures using this model is presented

The second main direction of the thesis, presented in chapter 5, is the use of expert systems in information fusion. In this application, three strategies are introduced to deal with this problem, one of them involving the use of the expert system. Measures of performance are made for each of them. Chapter 6 contains the results for the example and provides an interpretration of them. Conclusions and directions for future research are included in chapter 7.

16

CHAPTER 2

ANALYTICAL TOOLS

This chapter provides a description of the different tools used in this thesis.

2.1 INFORMATION THEORY

Information theory provides a theoretical framework for the analysis of workload in decisionmaking organizations. It allows to evaluate the activity of a decisionmaker by relating, in a quantatitative manner, the uncertainty in the tasks to be performed with the amount of information that must be processed to obtain certain results.

2.1.1 Entropy and Transmission

Two quantities defined in Information Theory, and which are essential for this purpose, are Entropy and Transmission. The entropy of the random variable x measures the uncertainty in the value x will take. The transmission between x and y represents the amount by which the knowledge of y reduces the uncertainty in x.

The entropy of a random variable x which takes values according to the probability distribution $p(x)$ is given by :

$$H(x) = - \sum_{x} p(x) \log_2(p(x)) \qquad (2.1)$$

The conditional entropy $H_x(y)$ is given by :

$$H_x(y) = - \sum_{x} p(x) \sum_{y} p(y|x) \log_2(p(y|x)) \qquad (2.2)$$

The joint entropy of x and y is given by :

$$H(x,y) = - \sum_{x} \sum_{y} p(x,y) \log_2(p(x,y)) \qquad (2.3)$$

The transmission or mutual information, between the variables x and y characterized by $p(x)$, $p(y)$ and $p(y|x)$ is given by :

$$T(x:y) = H(x) - H_y(x) = H(y) - H_x(y) = H(x) + H(y) - H(x,y) \qquad (2.4)$$

If x is deterministic and has no associated uncertainty, $H(x)$ is equal to zero. If x and y are independent, their mutual information $T(x:y)$, is zero. If y is a deterministic function of x, their mutual information is given by $H(y)$.

The generalization to n-dimensions is introduced to model information structures with multiple variables :

$$H(x_1, x_2, \ldots, x_n) = - \sum_{x_1 \ldots x_n} p(x_1, x_2, \ldots, x_n) \log_2(p(x_1, x_2, \ldots, x_n)) \qquad (2.5)$$

which can be expressed also by :

$$H(x_1, x_2, \ldots, x_n) = H(x_1) + H_{x_1}(x_2) + \ldots + H_{x_1, x_2, \ldots, x_{n-1}}(x_n) \qquad (2.6)$$

The transmission between n variables is given by :

$$T(x_1 : x_2 : \ldots : x_n) = \sum_{i=1}^{n} H(x_i) - H(x_1, x_2, \ldots, x_n) \qquad (2.7)$$

2.1.2 Partition Law of Information

If the set $\{x_1, x_2, \ldots, x_n\}$ is partitioned in disjoint subsets, the transmission of the n variables can be decomposed as follows:

$$T(x_1:x_2:\ldots:x_n) = T(x_1:x_2) + \ldots + T(x_{n-1}:x_n) + T(x_1,x_2:x_3,x_4:\ldots:x_{n-1},x_n) \quad (2.8)$$

The study of information processing has led to the introduction of the Partition Law of Information (Conant, 1976): the total information theoretic activity of a system is defined as the sum of the entropies of all the internal variables of the system. Furthermore, this total activity can be decomposed into four quantities that account for what actually occurs in the system. For an input variable x, N-1 internal variables w_i, i=1,...,N-1 and an output variable y also called w_N, the Partition Law of Information states :

$$\sum_{i=1}^{N} H(w_i) = T(x:y) + T_y(x:w_1, \ldots, w_{N-1}) + T(w_1:w_2: \ldots : w_{N-1}:y) + H_x(w_1, \ldots, w_{N-1},y)$$

$$(2.9)$$

$$G = G_t + G_b + G_c + G_n \quad (2.10)$$

The total activity of the system is denoted by G. The first term on the right-hand side, $T(x:y)$, is denoted by G_t : it is the throughput and represents the mutual information or transmission between the input and the output. The second term, $T_y(x:w_1,w_2,\ldots,w_{N-1})$, is denoted by G_b : it is the blockage of the system and represents the amount of information in the input that is not included in the output. The third term, $T(w_1:w_2: \ldots :w_{N-1}:y)$, is denoted by G_c : it is the coordination of the system and represents the amount by which all the internal variables in the system constrain each other. The last term, $H_x(w_1,w_2,\ldots,w_{N-1},y)$, is denoted by G_b : it is the noise of the system and represents the information internally generated.

2.2 PETRI NET THEORY

Petri Nets (Peterson, 1980, Reisig,1985) are used for the modeling and the analysis of concurrent and asynchronous processes. Their field of applications ranges from the modeling of manufacturing processes to the representation of the flow-charts of complex computer software. They have been succesfully used for the modeling of decisionmaking organizations (Remy et al. 1987) because they provide an explicit representation of the interactions among decisionmakers.

2.2.1 Definitions

A Petri Net - denoted by PN - is a bipartite directed graph represented by a quadruple
PN = (P, T, I, O) where :
P = {p_1, ..., p_n} is a finite set of n places,
T = {t_1, ..., t_m} is a finite set of m transitions,
I is a mapping PxT -> {0,1} corresponding to the set of directed arcs - called connectors - from places to transitions. I(p_i, t_j) = 1 means that there exists a connector from the place p_i to the transition t_j.
O is a mapping TxP -> {0,1} corresponding to the set of connectors from transitions to places.

An example of a Petri Net, PN1, is shown in Figure 2.1. Places are represented with circles and transitions with bars.
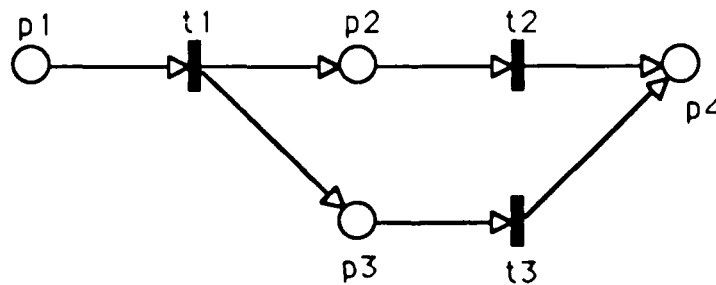
Figure 2.1 Petri Net PN1.

20

In this example, we have :

$$P = \{p_1, p_2, p_3, p_4\},$$

$$T = \{t_1, t_2, t_3\},$$

$I(p_1, t_1) = 1$	$I(p_2, t_1) = 0$	$I(p_3, t_1) = 0$	$I(p_4, t_1) = 0$
$I(p_1, t_2) = 0$	$I(p_2, t_2) = 1$	$I(p_3, t_2) = 0$	$I(p_4, t_2) = 0$
$I(p_1, t_3) = 0$	$I(p_2, t_3) = 0$	$I(p_3, t_3) = 1$	$I(p_4, t_3) = 0$

$O(t_1, p_1) = 0$	$O(t_2, p_1) = 0$	$O(t_3, p_1) = 0$
$O(t_1, p_2) = 1$	$O(t_2, p_2) = 0$	$O(t_3, p_2) = 0$
$O(t_1, p_3) = 1$	$O(t_2, p_3) = 0$	$O(t_3, p_3) = 0$
$O(t_1, p_4) = 0$	$O(t_2, p_4) = 1$	$O(t_3, p_4) = 1$

A **marking** of a Petri Net PN is a mapping $M : P \rightarrow \{0,1,2,...\}$ which assigns a non-negative integer number of tokens to each place of the net.

A transition t is **enabled** by a given marking M if and only if each of its input places contains at least one token, which means that :

$$\forall p \in P, M(p) \geq I(p,t). \qquad (2.11)$$

When a transition is enabled, it can **fire** : one token is removed from each input place and one is added to each output place. A new marking M' is reached given by the relation :

$$\forall p \in P, M'(p) = M(p) + O(t,p) - I(p,t). \qquad (2.12)$$

A Petri Net is pure if it has no self loop, i.e. no place is an input and an output of the same transition.

2.2.2 Linear Algebra

The structure of a pure Petri Net PN can be represented by an integer matrix C, called the incidence matrix of the Petri Net PN. Its elements are :

$$C_{ij} = O(t_j, p_i) - I(p_i, t_j), \qquad 1 \le i \le n, 1 \le j \le m \qquad (2.13)$$

C_{ij} only takes the value 0, 1 and -1. The incidence matrix of PN1 is :

$$C(PN1) = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}$$

The incidence matrix of a Petri Net is used to determine the simple paths of the net which are the paths (place-transition- ...-transition-place) linking the entry place of the net (called the source) to the exit place (called the sink). Simple paths show the sequence of steps for the modeled process. The incidence matrix is also useful for the determination of the slices of the net, which are the sets of places or transitions which represent concurrent activity in the process modeled by the Petri Net. In the example shown on Figure 2.1, the source of the net is place p_1, the sink is place p_4. The simple paths are :

simple path 1 : $p_1 - t_1 - p_2 - t_2 - p_4$.
simple path 2 : $p_1 - t_1 - p_3 - t_3 - p_4$.

The slices are :

slice 1 : $\{p_2, p_3\}$.
slice 2 : $\{t_2, t_3\}$.

Details on the determination of the simple paths and of the slices of a Petri Net can be found in Hillion (1986) and in Jin et al. (1986).

2.2.3 Petri Nets with Switches

For the modeling of decision making organizations, switches have been introduced as an extension of the Petri Net theory to take into account the possibility of alternatives (Tabak and Levis, 1985). A switch is a particular transition with multiple output places. When a switch fires only one of its output places can receive a token. This output place which receives the token is chosen according to certain decision rules associated with the switch. These decision rules can be anything : they can be deterministic (the output place is a function of the input), or stochastic (a probability distribution over the set of possible output places is defined). Figure 2.2 depicts a Petri Net, PN2, with a switch S1.
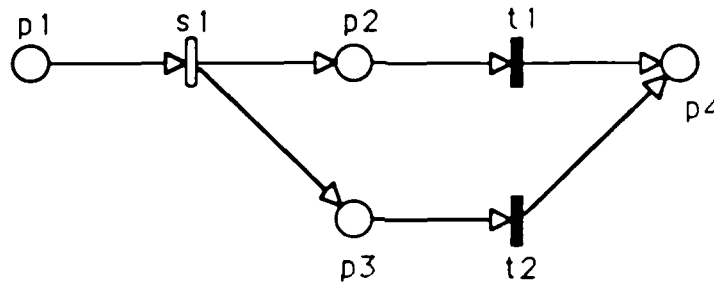
Figure 2.2 Petri Net PN2 with a switch

A pure Petri Net with switches can be represented also with an incidence matrix. Switches are considered to be transitions and appear at the last columns of the matrix. Nothing about the decision rules of the switch is contained in the matrix representation. The incidence matrix of the Petri net PN2 is :

$$
C(PN2) = \begin{array}{ccc} t1 & t2 & s1 \end{array} \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & 0 \end{bmatrix}
$$

23

For the modeling of decisionmaking organizations, a Petri Net is a formal model of information flow. Tokens can be considered as symbolic information carriers ; places are the nodes where tokens can stand without being modified ; transitions and switches are events that perform a transformation on the information : it can be a transmission, a computation or a decision ; switches are particular kinds of events that transform input information according to a certain decision rule.

2.3 PREDICATE TRANSITION NETS

Petri Nets have been shown to be insufficient to model clearly large processes or a set of different processes which use the same resources. The modeling tool needs to be extended to allow the handling of different classes of tokens. There exist different extensions of Petri Net theory which allow the manipulation of different kind of tokens : coloured Petri nets (Jensen, 1981), and Predicate Transition Nets (Genrich and Lautenbach, 1981) are the best known. In this thesis, an adapted model of Predicate Transition Net is used in chapter 4 for the modeling of fuzzy logic operators used in expert systems.

Predicate Transition Nets have the following characteristics.

2.3.1 Tokens

Each **token** travelling through the net has an identity and is considered to be an individual of a given class called variable. Each variable can receive different names. For example, we say that a variable represented by x takes the values a, b, or c. The set of the values x can take is represented by :

$$\underline{x} = \{a, b, c\}$$

Another name for the variable represented by x which can take the same values is y. We have :

$$\underline{y} = \underline{x} = \{a, b, c\}$$

24

A token is therefore an occurence of x (or y) having taken a value and is represented formally by $x|_{x=a}$. It is important to note that the token $y|_{y=a}$ is a different instance of the token $x|_{x=a}$. For simplicity, in the figures, a token are represented by its value : a instead of $x|_{x=a}$, for example.

2.3.2 Places

Places are entities which can contain tokens before the firing of transitions. A place can contain token from different classes.

The **marking** of a place is a formal sum of the individual tokens contained in the place. For example, if we suppose that {a,b,c} is a class of tokens and {<a>, <b>, <c>} is another class of tokens, a place P containing the tokens a, a, b and <c> has the marking M(P) :

$$M(P) = 2a + b + <c>$$

The marking of a Predicate Transition Net with n places is a n-dimensional vector with elements indicating the marking of each place.

2.3.3 Connectors and Labels

Each **connector** has a label associated with it which indicates the kinds of tokens it can carry. A special grammar is used on the labels to define in what way tokens can be carried. The labels of connectors linking places to transitions are conditions to fulfill to carry the tokens. The labels of connectors linking transitions to places indicate what kind of token to place in the places after the firing of the transition.

If we consider that :

    {a,b,c} is a class of tokens of which the name can be x and/or y ,
    {<a>, <b>, <c> } is another class of token of which the name can be <z>,

the following notation in labels is used :

When names are joined by the sign "+" then the tokens defined by these names have to

be carried at the same time. For example, the label "x + y + <z>" indicates that two tokens of the first class and one token of the second have to be carried together at the same time by the connector.

When names joined by the sign "," then the tokens defined by these names can be carried at different times but not together. For example, the label "x, <z>" indicates that either a token of the first class or a token of the second can be carried.

Mixing of notation is possible. The label "(x+y),<z>" indicates that the connector can carry either two tokens of the first class or one token of the second.

A connector without label has no constraint on the kind of tokens it can carry.

2.3.4 Transitions

Transitions have attached to them a predicate which is a logical formula (or an algorithm) built from the operations and relations on variables and tokens in the labels of the input connectors. The value (true or false) taken by the predicate of a transition depends on the tokens contained in the input places of the transition. When the predicate has the value "true", the transition is enabled and can fire.

A transition without predicates is enabled as soon as all the input places contain the tokens specified by the labels of the connectors.

Transitions with predicates are represented graphically with squares or rectangles. The predicate is written inside. Transitions without predicates are represented with bars as in the ordinary Petri Nets.

2.3.5 Firing Process

The conditions of enabling of a transition are (1) the input places contain the combination of tokens specified by the labels of the connectors and (2) the predicate of the transition is true. If these two conditions are fulfilled, the transition can fire. In the firing process, tokens specified by the input connectors are withdrawn from the corresponding input places and tokens specified by the output connectors are put in the output places.

Let us consider the following examples to illustrate this process. We assume that there are three classes of tokens :

{1, 2, 3} represented by x and/or y and/or z,

{<1>, <2>, <3> } represented by <z>.

{<1,3>, <2,3> } represented by <x,y> and/or <x,z>.

In example 1, shown on Figure 2.3, the right tokens required by the label of the connectors are present in the input places. However, the predicate of the transition is false. Therefore the transition is not enabled and cannot fire.
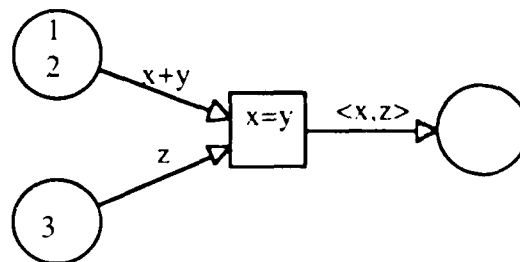
Figure 2.3 Predicate transition net : example 1

In example 2, shown in Figure 2.4, the transition is enabled : the right combination of tokens specified by the connectors are in the input places and the predicate is true. The transition can fire.

In the firing process the tokens of the input places which participated in the enabling of the transition are withdrawn. The token <1, 3> specified by the label of the output connector is put in the output place.

27
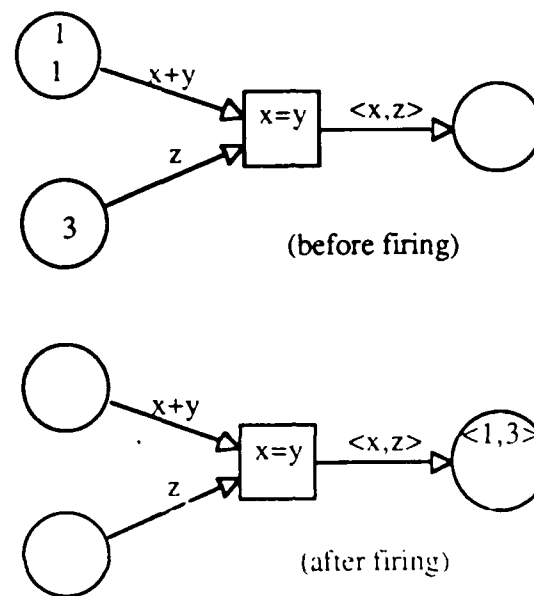
Figure 2.4 Predicate transition net : example 2

Example 3, shown in the Figure 2.5. illustrates how two different classes of tokens can be processed. The label "<x>,x" indicates that the input place has to contain either a token of the first class or a token of the second. Both are present but it is important to note that the x used in the two parts of the label does not require any relation between the two kinds of tokens but refers to the name of the variable used in the predicate of the transition. The transition is enabled. The variable x used in the predicate is equal to 1 because the token <1> in the first input place allows the predicate to be true and the token 3 does not.

In the firing process, only the tokens involved in the enabling are withdrawn from the input places : <1> from the first, 2 from the second. The token <1,2> of the third class is put in the output place.

In this last example, if there had been a token 1 instead of a token 3 in the first input place, a conflict on the choice of the token to withdraw in the firing would have occured. Different strategies for the conflict resolution exist : random choice, choice made by the user, ... In this thesis, these conflicts are avoided by limiting the number of classes and the capacity of the places.
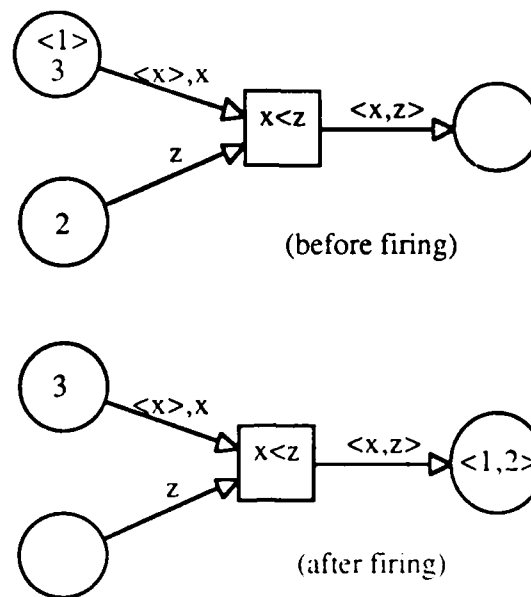
Figure 2.5 Predicate transition net : example 3

2.4 MODEL OF THE INTERACTING DECISIONMAKER

2.4.1 The Basic Model

The model of a decisionmaker interacting with an organization introduced by Boettcher and Levis (1982-a) is shown in Figure 2.6. The model consists of four stages :

In the **situation assessment** (SA) stage, the decisionmaker receives an input X from the environment and processes this information through the use of one of the U algorithms $f_1,...,f_U$. The choice of a particular algorithm is done by the decision switch on the basis of a decision rule which does not necessarily depend on the input X. The result of the SA is Z. The decisionmaker can also transmit some information, Z^{io}, to the rest of the organization.

In the **information fusion** (IF) stage, the decisionmaker can merge his own situation assessment, Z, with some information, Z^{oi}, communicated by the rest of the organization. The result of this stage is Z'.

29

In the **command interpretation** (CI) stage, the decisionmaker can receive
V^{oi}, from decisionmakers hierarchically superior to him which modify the basis for selecting
a final response Y.

In the **response selection** (RS) stage, the decisionmaker selects a response Y by
processing the input Z' through the use of one of the V algorithms $h_1 \ldots h_V$. This choice is
represented by the second switch in the Figure 2.6. The rule of this switch is affected by the
command V^{oi}.
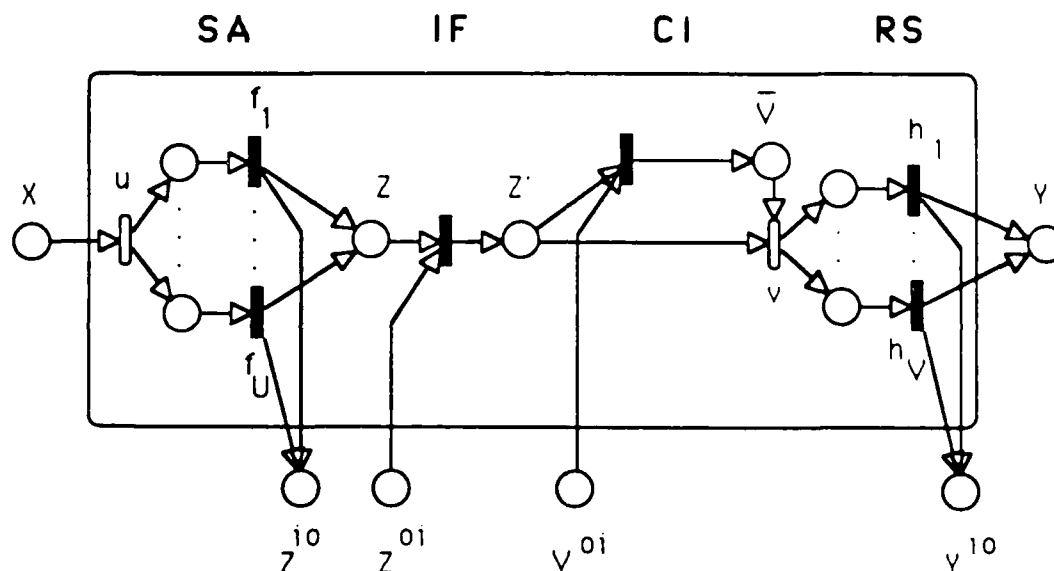
Figure 2.6 Model of the interacting decisionmaker.

A pure decision strategy for the decisionmaker is one for which both the situation
assessment strategy p(u) and the response selection strategy p(v | Z) are pure, i.e., one of the
algorithms f_i is selected with probability 1 and one of the algorithms of the response selection
is selected with probability 1. The kth pure strategy of the decisionmaker is therefore :

$$D_k = \{p(u=i) = 1, p(v = j | Z) = 1\} \qquad (2.14)$$

30

The maximal number of pure strategies for the decisionmaker is equal to UV^M, where U, V, and M are respectively the number of algorithms in the situation assessment stage, the number of algorithms in the response selection stage and the number of values the variable Z can take. The other internal decision strategies are mixed and are obtained as a convex combination of the pure strategies.

Because of the possible interactions among the decisionmakers, measures of performance (described in the next section) depend on the strategy employed by the organization. In an organization with n decisionmakers, a pure organizational strategy is defined as a n-tuplet of pure strategies (one for each decisionmaker) :

$$\Delta_{k_1, k_2 \dots k_n} = (D^1_{k_1}, D^2_{k_2}, \dots, D^n_{k_n}) \tag{2.15}$$

where $D_{k_i}^i$ is the k_i th pure decision strategy of the decisionmaker i, as expressed by equation (2.14). Finally, since each decisionmaker is assumed to select his strategy independently of the rest of the organization, the behavioral organization strategy is defined as a n-tuplet of mixed strategies (one for each decisionmaker).

2.4.2 Measures of Performance of an Organization

The decisionmaking organization (DMO) can be considered as a system which must perform a certain task or mission. We assume that the task of the DMO is to process an input X taking values in a finite alphabet A with a discrete probability distribution $p(X)$ and that a cost $C(Y)$ exists for every response Y of the organization. The real probability that the DMO receives the input X_i is $p(X=X_i)$. One possible way to define $C(Y)$ is to map X_i into an ideal response Y_{di} and then assign a cost $C(Y,Y_{di})$ to the difference between the actual organization response Y and the desired response Y_{di}. $p(X)$ symbolizes the fact that all the inputs are not equally likely. In the same way, $C(Y)$ symbolizes the fact that the processing of different inputs may have different utilities or interests for the organization designer.

In this context, the measures of performance (MOP) of the system and of the mission are accuracy, timeliness, and workload. It is possible to assess the organization's effectiveness in performing its task by measuring :

31

The **accuracy** of its response, denoted by J, which shows how well its response correspond to the desired responses (Andreadakis and Levis, 1987). For each input X_i and each organization strategy j, the response Y_j is produced and the cost $C(Y_j, Y_{di})$ is computed. The accuracy of the organization is then defined as follows :

$$J = \sum_i p(X_i) \sum_j C(Y_j, Y_{di}) p(Y_j | X_i) \tag{2.16}$$

The **timeliness** of its response, T, i.e., the extent to which those responses are provided at the right times. It is possible to define several measures of timeliness : it can be the expected response time or the probability that the response time lies inside an interval $[T_{min}, T_{max}]$.

The **workload** G^i of decisionmaker DM_i, i.e., his processing activity in carrying out his part of the task. It is obtained by computing the entropy of all of his internal variables in accordance with the Partition Law of Information.

The mission requirements are specified in terms of constraints on the values that those MOPs can take. The bounded rationality constraint (Boettcher and Levis, 1982a and 1982b) expresses that G^i verifies : $G^i \leq F^i.\tau$, where F^i characterizes the maximum activity rate of the decisionmaker DM_i and τ, the mean input interarrival time. Similarly, for accuracy, the notion of satisficing is expressed as $J \leq J_0$.

2.5 CONCLUSION

Several tools have been decribed in this chapter. They are used in the different parts of this thesis. Information theory is useful for quantifying the workload of a decisionmaker. Petri Net theory is applied to the representation of decisionmaking organizations using the model of the interacting decisionmaker shown in section 2.4. One of the advantages of the Petri Net formalism is that it allows to obtain time-related measures of these organizations. Finally, Predicate Transition Nets provide a useful extension of the ordinary Petri Nets. They allow the handling of different classes of tokens. Therefore, concurrent and asynchronous

processes can be represented in more detail. They are used later in this thesis to model fuzzy logic operators and to represent the dynamical behavior of an expert system. The next chapter provides a description of the Knowledge Based Expert System technology. It describes different ways to deal with uncertainty, especially the fuzzy logic formalism.

CHAPTER 3

DEALING WITH UNCERTAINTY IN EXPERT SYSTEMS

3.1 EXPERT SYSTEMS

3.1.1 Structure of an Expert System

Knowledge Based Expert Systems, commonly called Expert Systems, are - in theory - able to reason using an approach similar to the one followed by an expert when he solves a problem within his field of expertise. An expert system can be used for many purposes : to control, to diagnose, to solve problems, to plan, to design, ...

There are three distinct components in an expert system (Figure 3.1) :

(1) The **knowledge base** which contains the set of information specific to the field of expertise. Knowledge is expressed in a language defined by the expert. The knowledge base is a collection of general facts, rules of thumb and causal models of the problem domain. A number of formalisms exist to represent knowledge. The most widely used is the production system model in which the knowledge is encoded in the form of antecedent-consequent pairs or IF-THEN rules.

(2) The **fact base**, also known as context or working memory, contains the data for the specific problem to be solved. It is a workspace for the problem constructed by the inference mechanism from the information provided by the user and the knowledge base. The working memory contains a trace of every line of reasoning previously used by memorizing all the intermediate results. This can therefore be used to explain the origin of the information deduced or to describe the behavior of the system.

(3) The **Inference Engine** is used to monitor the execution of the program by using the knowledge base to modify the context. It uses the knowledge and the heuristics contained in the knowledge base to solve the problem specified by the data contained in the fact base. Often, the knowledge base is composed of rules of the

kind, A -> B, saying that if A is valid, B can be deduced. In this case, the inference engine selects, validates and triggers some of these rules to reach the solution of the problem.

```
┌─────────────────┐
│                 │
│   FACT BASE     │ ◁──────┐        ┌──────────────┐
│   (Context)     │        │        │  INFERENCE   │
│                 │        └───────▷│   ENGINE     │
└─────────────────┘                 │    find      │
                                    │    select    │
┌─────────────────┐        ┌───────▷│    execute   │
│                 │        │        └──────────────┘
│  KNOWLEDGE BASE │ ◁──────┘
│                 │
└─────────────────┘
```
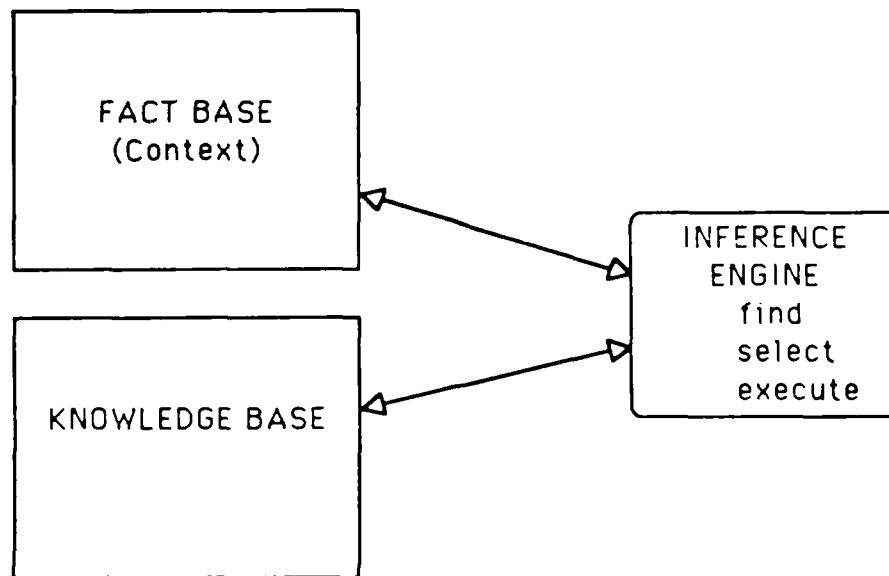
Figure 3.1 Structure of an Expert System

Among the strategies used by the inference engine to select the rules, forward chaining and backward chaining are the most common. In **forward chaining**, the inference mechanism works from an initial state state of known facts to a goal state. It finds first all the rules that match the context, it then selects one rule based on some conflict resolution strategy, and then execute the selected rule. Facts are inputs to the system. The most appropriate hypothesis that fits the facts is deduced. For **backward chaining**, the system tries to support a hypothesis by checking known facts in the context. If these known facts do not support the hypothesis, the preconditions needed for the hypothesis are set up as subgoals. The process for finding a solution is to search from the goal to the initial state and involves therefore a depth-first search. The model described in Chapter 4 uses backward chaining as the strategy for finding the solution.

36

3.1.2 Inference Net

The relationships among the rules of a production system can be represented with an inference net. It allows to show graphically the logical articulation of different facts or subgoals, and to identify which rules are used to reach a specific goal. Let us consider the following production rules :

> if A AND B, then C
> if D OR E, then F
> if NOT G, then H.

These rules are represented in the inference net formalism on Figure 3.2.
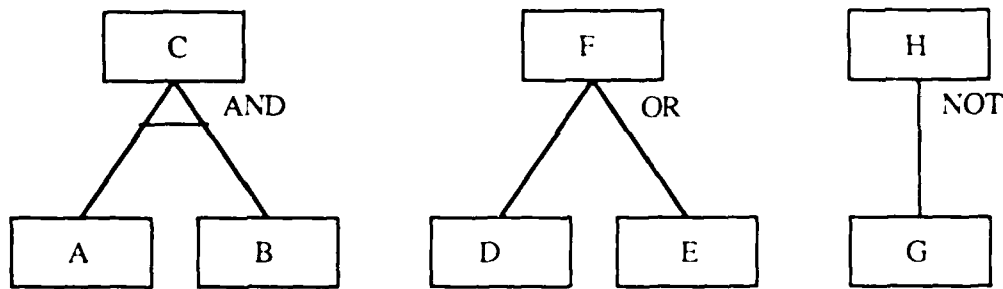
Figure 3.2 Representation of the logical operators in the inference net formalism

3.1.3 Consultant Expert Systems

We will focus our effort on the most common kind of expert system : the **consultant expert system**, as described by Johnson and Keravnov (1985). Most systems engage in a dialogue with the user, the computer acting as a "consultant," by suggesting options on the basis of its knowledge and the symbolic data supplied by the user. The dialogue terminates when a decision or a recommendation is reached. The formalism used to represent knowledge in consultant expert systems is the production system model described earlier in this chapter.

Moving from known items of information to unknown information is the vital process of a consultant system. The user of a consultant expert system has "observed" some particular state of affairs within the domain of the system's expertise and submits these observations to the system. Examples of these states are a sick person, a faulty machine and a malfunctioning business environment. Based on the observations, the system makes inferences and suggests new routes of investigation which will yield high grade information. Interactions continue until the system finds the most likely explanation of the observations.

The mode of interaction used most often by these kinds of expert systems is the computer initiated mode (as opposed to the user initiated mode) where the user is restricted to responding to system's requests only.

3.2 DEALING WITH UNCERTAINTY

One aspect of great interest in expert systems is their ability to handle symbolic objects which have a certain level of uncertainty. This is very important because real life problem solving requires the acceptance of uncertainty in order to minimize the difficulty of the problem. Various methods have been tried, some quite successfully, which allow the use of fragmentary and uncertain information to reach an estimate of the truth. The application of this ability to handle uncertainty in the fusion of inconsistent data is presented in section 5.4.

Since the handling of symbolic objects is subject to uncertainty, expert system research has been accompanied by the development of methods for being precise about imprecision. There are different ways of reasoning under uncertainty where the problem data and/or the rules of inference are not 100% reliable.

Forsyth (1984) shows three main approaches to deal with this problem which have been used quite successfully in different systems :

1- **Certainty Factors** are a scheme that has been successfully used in MYCIN by Buchanan and Shortliffe (1984). Certainty factors measure the confidence that can be placed in any given conclusion as a result of the preceding evidence. A certainty factor is defined as the difference between two component measures :

$$CF[h : e] = MB[h : e] - MD[h : e] \qquad (3.1)$$

where :

CF[h : e] is the certainty factor of the hypothesis h given by the evidence e.

MB[h : e] is the measure of belief in h given e.

MD[h : e] is the measure of disbelief in h given e.

CF can range from -1 (completely false) to +1 (completely true) with values in between, 0 representing ignorance. This definition is easily extended to more than one piece of evidence :

$$MB[h : e1, e2] = MB[h : e1] + MB[h : e2] - MB[h : e1]\ MB[h : e2] \qquad (3.2)$$

Furthermore, this model can include the possibility that inference may be uncertain as well as data. Each rule has therefore an attenuation number between 0 and 1 which indicates its reliability.

2- **BAYES rule** (Duda, Hart and Nilsson, 1984) is used to tie together information from disparate sources. It provides for computation of relative likelihoods between competing hypotheses on the strength of evidence. The likelihood ratio is defined as the ratio of the probability of the event or evidence E given a particular hypothesis H over the probability of the evidence given the falsity of that hypothesis (not H). It is given by the formula :

$$LR(H : E) = \frac{P(E \mid H)}{P(E \mid \text{not } H)} \qquad (3.3)$$

Thus, if we know the probability of the evidence given the hypothesis and its negation, we can determine the likelihood of the evidence given the hypothesis in the light of the evidence. LR > 1 indicates the evidence is favorable to the hypothesis. If LR < 1, the evidence is contra-indicative. If LR = 1, it is neutral.

The likelihood ratio can be used to adjust the odds in favour of the hypothesis in question, if the evidence has occurred. Odds in favour (F) is defined with the probability of certitude P by the formula :

$$F = \frac{P}{1-P} \qquad (3.4)$$

The entire scheme is summarized in the expression :

$$F'(H) = F(H) \ LR(H : E) \qquad (3.5)$$

where :

F(H) is the prior odds in favour of H

F'(H) is the resulting posterior odds given the event E as determined by the likelihood ratio.

Finally the likelihood ratio can be adjusted, if the evidence is itself uncertain by computing a scaled ratio LR' such that :

$$LR' = LR \ P(E) + (1 - P(E)) \qquad (3.6)$$

where P(E) is the probability that the evidence is valid.

3 - **Fuzzy Logic** (Zadeh, 1983) deals with uncertainty in more general way than the methods previously described. In these methods, the assumption of conditional independence of hypotheses is made and, consequently, these approaches are not very useful when this assumption is relaxed. The Fuzzy Logic approach seems appropriate. This method has been successfully used as the underlying inference mechanism in different production systems (Whalen and Schott, 1983) and particularly in the decision support system REVEAL (Jones and Morton, 1982). This is the method chosen in this thesis to deal with the problem of uncertainty .

3.3 FUZZY SET THEORY AND FUZZY LOGIC

3.3.1 Fuzzy Set Theory

A fuzzy set (Zadeh, 1965) is defined as follows :

Let $X = \{ x \}$ be a collection (a set) of objects denoted generically by x. A fuzzy set A in X is a set of ordered pairs :

$$A = \{ (x, \mu_A(x)) \} \quad x \in X \qquad (3.7)$$

where $\mu_A(x)$ is the grade of membership of x in A. μ_A is a function from X to M, the membership space. Often, M is the interval [0,1] and will be used as such in this thesis.

This formulation allows one to handle concepts or things which are fuzzy in nature and can not always be described with numbers. For example, let X be a collection of persons, $X = \{$John, Steve, Charles$\}$ and we know that :

> John is 25 years old,
> Steve is quite old,
> Charles is very old.

If A denotes the fuzzy set of old persons in X, A will be :

$$A = \{ (John, 0.3), (Steve, 0.7), (Charles, 1) \}$$

The numbers (0.3, 0.7, 1) show to what extent the persons belong to the fuzzy set A. We can see that fuzzy concepts like *quite old* or *very old* which are not attached to any real numerical values can be taken into consideration. It is important to note also that these grades of membership to the set A are defined in a subjective way. They can, for example, depend on the environment in which the set X is defined. If X defined above is a collection of students, the fact that Charles is very old does not necessarily mean that Charles is very old in the collection of all the inhabitants of a town.

In addition, for this example, when the age is known, the mapping function giving the degree of oldness (or the grade of membership to the fuzzy set of old persons) from the age of a person is also defined in a subjective way. There are no strong rational grounds for preferring one mapping function to another.

3.3.2 Operations on Fuzzy Sets and Fuzzy Logic.

The theory of fuzzy sets offers also a collection of operations on the sets :

equality : $\qquad A = B$ if $\mu_A(x) = \mu_B(x), \quad \forall\, x \in X.$ $\qquad$ (3.8)

inclusion : $\qquad A \supset B$ if $\mu_A(x) > \mu_B(x), \quad \forall\, x \in X.$ $\qquad$ (3.9)

complementarity : A' is the complement of the fuzzy set A if :
$$\mu_{A'}(x) = 1\text{-}\mu_A(x), \quad \forall\, x \in X. \qquad (3.10)$$

intersection : $\qquad \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), \quad \forall\, x \in X. \qquad$ (3.11)

union : $\qquad \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), \quad \forall\, x \in X. \qquad$ (3.12)

The three last operations are the basis for the definition of fuzzy logic. Fuzzy logic operates on numbers between 0 and 1 which indicate partial truth denoted by p and which are equivalent to the grades of membership in different fuzzy sets. For example, the partial truth of the statement "Paul is tall" will be equal to the grade of membership of Paul in the fuzzy set of tall persons. The equivalents of the AND, OR and NOT operators are defined as follows :

$$
\begin{aligned}
&p1 \text{ AND } p2 = \min(p1, p2) \\
&p1 \text{ OR } p2 = \max(p1, p2) \qquad\qquad (3.13) \\
&\text{NOT } p1 = 1 - p1
\end{aligned}
$$

Thus, pieces of evidence are combined in a rigorous and consistent manner by computing the resulting partial truth or grade of membership from the partial truths or grades of membership of the pieces of evidence. For example, the partial truth of the statement "Paul is not tall" will be equal to the grade of membership to the fuzzy set which is complementary

to the fuzzy set of tall persons. In the same way, the partial truth of the statement "Paul is tall and fat" will be measured by the grade of membership of Paul in the intersection of the fuzzy set of tall persons and of the fuzzy set of fat persons, and will be equal to the minimum of the grades of membership in these tv o sets.

3.3.3 Interest for Artificial Intelligence.

In spite of this problem of subjectivity in the definition of these grades of membership, the interest for Artificial Intelligence is very significant. Indeed, the problems to be solved in AI rarely involve the distinction between two opposed elements. As Zadeh (1983) points out, with two valued logic (true-false), all classes are assumed to have sharply defined boundaries. So either an object is a member of class or it is not a member. But in the real world, most classes do not have sharp boundaries. This is the case for characteristics or properties like tall, intelligent, tired, sick... Two-valued logic is not designed to deal with properties that are a matter of degree. Fuzzy logic allows this and opens new horizons. It underlies inexact or approximate reasoning. Fuzzy logic reduces the gap between the imprecise human reasoning and the reasoning used by computer. It allows the processing of unclear knowledge and common sense reasoning.

Another great interest of Fuzzy logic is that it is a formalism which allows for and encourages the use of natural language as its means of interaction with the user. We have presented earlier in this chapter how fuzzy sets and fuzzy logic were able to deal with and process natural language quantifiers (quite, very, ...) in a structured and consistent way.

However, this approach for dealing with the fuzziness of the real world does not solve everything. Forsyth (1984) shows that one of the most important problems is the weighting of disparate pieces of evidence. How does one resolve the conflict of two different contradictory truth values for the same proposition obtained with two different sets of pieces of evidence ? Do we take the minimum, the maximum, the average or some other function of the two numbers ? Let us consider the following rules :

"John is tall" if "John plays basketball"   (1)
"John is tall" if "John is more than 6 feet tall"   (2)

If we know that John plays basketball but is 5'11" tall, the rules (1) and (2) will lead to completely different partial truth for the statement "John is tall". The problem is to define which final value to assign to this statement. There is no right answer. One way to solve this problem is to assign some degree of likelihood for each rule which leads to the same conclusion. In the previous example, the rule (2) will have a higher degree of likelihood and would give the final partial truth of the statement. The problem is about the same for a proposition of which the different truth values are obtained through different rules pointing in the same direction, tending to confirm each other and reinforcing belief in the conclusion. The resulting truth value should indicate a higher degree of truth than the average or even the maximum of the obtained truth value.

CHAPTER 4

AN EXPERT SYSTEM MODEL USING PREDICATE TRANSITION NETS

4.1 STRUCTURE OF THE EXPERT SYSTEM

4.1.1 The Knowledge Base

The model developed in this thesis permits the explicit representation of the rules of a knowledge base and the relationships among them. The kind of rule we are considering here is the production rule. A production rule is divided in two parts :

A set of conditions (called left-hand side of the rule) combined logically together with a AND or a OR operator,

A set of consequences or actions (called also right-hand side of the rule), the value of which is computed according to the conditions of the rule. These consequences can be the conditions for other rules. The logical combination of the conditions on the left-hand side of the rule has to be true in order to validate the consequences and the actions.

An example of a production rule is :

    IF   the flying object has delta wings AND
            the object flies at great speed
    THEN the flying object is a fighter plane.

The conditions "the flying object has delta wings" and "the object flies at a great speed" have to be true to attribute the value true the consequence "the flying object is a fighter plane."

4.1.2 The Inference Engine.

Logic used : In order to deal with uncertainty in items of evidence, fuzzy logic is implemented in the model to combine logically the conditions of the left-hand side of the rules. The value of a rule or a fact is either unknown or a number between 0 and 1, representing the degree of truth associated with it. The operators AND, OR, and NOT execute operations on these degrees of truth (see section 3.3.2).

$$p1 \text{ AND } p2 = \min(p1,p2)$$
$$p1 \text{ OR } p2 = \max(p1,p2) \qquad (4.1)$$
$$\text{NOT } p1 = 1\text{-}p1.$$

Process of Selection and Firing of Rules : In order to simulate the behavior of an expert system, the process of selection and firing of rules done by the inference engine has been modeled when a backward chaining strategy is used. A trigger is associated with every rule (or operator). A rule is selected by the inference engine when the trigger is activated. Only one rule at a time can be activated and the continuation of the selection and firing process is done according to the result of the rule :

If the result is unknown, the rule is put in memory and the rule which gives the value of the first unknown precondition is selected.

If the result is known, the last rule which was put in memory is selected again because the produced result is the value of one of its preconditions.

Let us consider the example where we have two rules :

$$B \Rightarrow C \qquad (1)$$

$$A \Rightarrow B \qquad (2)$$

and where the degree of truth of the fact A is known.

The inference engine selects first the rule (1). The degree of truth of C is unknown because the degree of truth of B is unknown. The rule (1) is then de-activated and put in

memory. The rule (2) is selected. Since the value of A is known, the value of B is deduced. Rule (1), which is the last to have been put in memory, is selected again and the answer C is given.

Search for efficiency : The process of selection and firing of rules described above is repeated by recursion until the final answer is found ; the process can last a long time. In the search for efficiency and performance, unnecessary computations must be avoided. In some cases, there is no need to know the values of all the preconditions of a rule to deduce the value of its consequence. For example, in boolean logic, if we have the rule :

$$A \text{ AND } B => C.$$

and we know that :

$$A \text{ is false,}$$

then the consequence C is false and there is no need to look for the value of B to conclude that ; the set of rules giving the value of B can be pruned.

In systems using fuzzy logic, this avoidance of unnecessary computations is all the more important as computations are more costly in time and memory storage than in systems using boolean logic. The problem is that little improvement in performance is obtained if extra computation is avoided only in the case of complete truth (for the operator OR) or of complete falsity (for the operator AND). The solution lies in the setting of thresholds for certain truth and certain falsity. For example, in the case of the operator AND, if we have :

$$A \text{ AND } B => C$$

and we know that the degree of truth of A is less than the threshold of certain falsity, we can deduce that the degree of truth of the consequence C is less than the degree of truth of A, and therefore, less than the threshold of certain falsity. There is no need to know the degree of truth of the precondition B. The thresholds for which no further search is required in the execution of the operators are set to 0.8 for certain truth in the operator OR and 0.2 for certain falsity in the operator AND. A rule or fact having a degree of truth larger or equal to 0.8 (resp. less or equal to 0.2) will be considered to be true (resp. false). Therefore, the logic takes into account the unknown rules or facts.

4.1.3 The Fact Base

Since backward chaining is used, it is important to store the result of every rule scanned by the system because it can be used several times by different operators and can be used to explain how the answer has been found.

4.2 CHARACTERISTICS OF THE PREDICATE TRANSITION NETS USED IN THE MODEL

Predicate Transition Nets described in section 2.3. are used to represent the expert system with fuzzy logic.

4.2.1 Classes of Tokens

Two classes of tokens are differentiated :

(1) The first one is the set of the real numbers between 0 and 1, representing the degrees of truth of the facts or items of evidence and is denoted by P. The names of the individual tokens of these classes will be p, $p1$, $p2$.

(2) The second class is denoted by S. The individuals of this class can only take one value. Only one token of this class will travel through the net and will represent the action of the inference engine in triggering the different rules.

4.2.2. Places.

Three kinds of places are differentiated :

(1) places representing a fact or the result of a rule and containing tokens of the class P or no token at all,

(2) places used by the system as triggers of operators and containing the token of the class S. These places and the connectors connected to these places are represented in bold style in the Figures and constitute what is called in this chapter the **system net**.

(3) places allowed to contain different kinds of tokens (P and S) and which are used to collect the tokens necessary for the enabling of the transitions of which they are the input places.

4.2.3 Connectors

The capacity of each connector is determined by its label. The label will define the class of tokens it can carry. The grammar of the label defined in section 2.3 is used : the label "p+S" indicates that the connector can carry a token of the class P and a token of the class S at the same time.

In some cases, the connector has to carry the token of class S when there is no token of the class P involved in the firing of a transition. The statement "absence of token of the class P" is denoted by the symbol Ø. This symbol is used in the labels, as if it was a class of tokens, in association with the names of the other classes. The symbol Ø is used in the following cases :

(1) The label "S+Ø" means that the connector can carry a token of the class S, if there is no token of the class P.

(2) The label "(S+p),(S+Ø) means that the token can carry, either a token of the class S and a token of the class P, or a token of the class S, if there is no token of the class P.

4.2.4. Transitions

In order to be enabled, conditions on the variables of the class P must be fulfilled. A transition will therefore be enabled, if the input places contain the number and the classes of tokens defined by the label of the connectors linking those places to the transition, and if the conditions on the variables are fulfilled. Let us consider the example shown on Figure 4.1 :
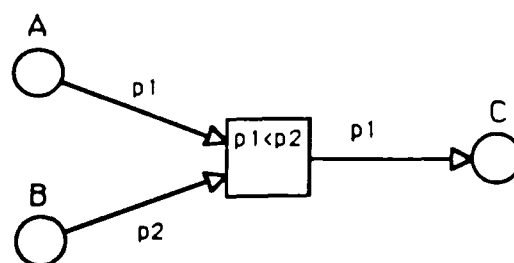
Figure 4.1. Example of a transition with a predicate

The condition "p1 < p2" written is the transition represented by a square is true when the value of the token named p1 coming from place A is less than the value of the token named p2 coming from place B, as specified by the connectors.

4.3 LOGICAL OPERATOR MODELS

The models of the logical operators AND, OR, and NOT are shown in Figures 4.2, 4.3 and 4.4. Let us describe now what happens in the operator AND (the operators OR and NOT behave in a similar way).

The operator drawn in Figure 4.2 realizes the operation :

$$A \text{ AND } B \Rightarrow C.$$

It can be represented as a black box, having three inputs : A, B and S_C (the trigger) and six outputs : C (the result), A, B (memorizing of the input value) and three system places S_A, S_B and S_{next}. Only one of those system places (represented in bold style in the figures) can have a system token at the output. S_{next} will contain a system token if the result of the operation is known, i.e., if C contains a token of the class P. This shows that the next operation can be performed. If the result is unknown, i.e., the two inputs are not sufficient to yield a result, the system token is assigned in order to get the values of these unknown inputs. Therefore, a system token will be assigned to S_A if (1) C is unknown and (2) A is unknown or if A and B are both unknown. The system token will be assigned to S_B if C is unknown and only B is unknown.
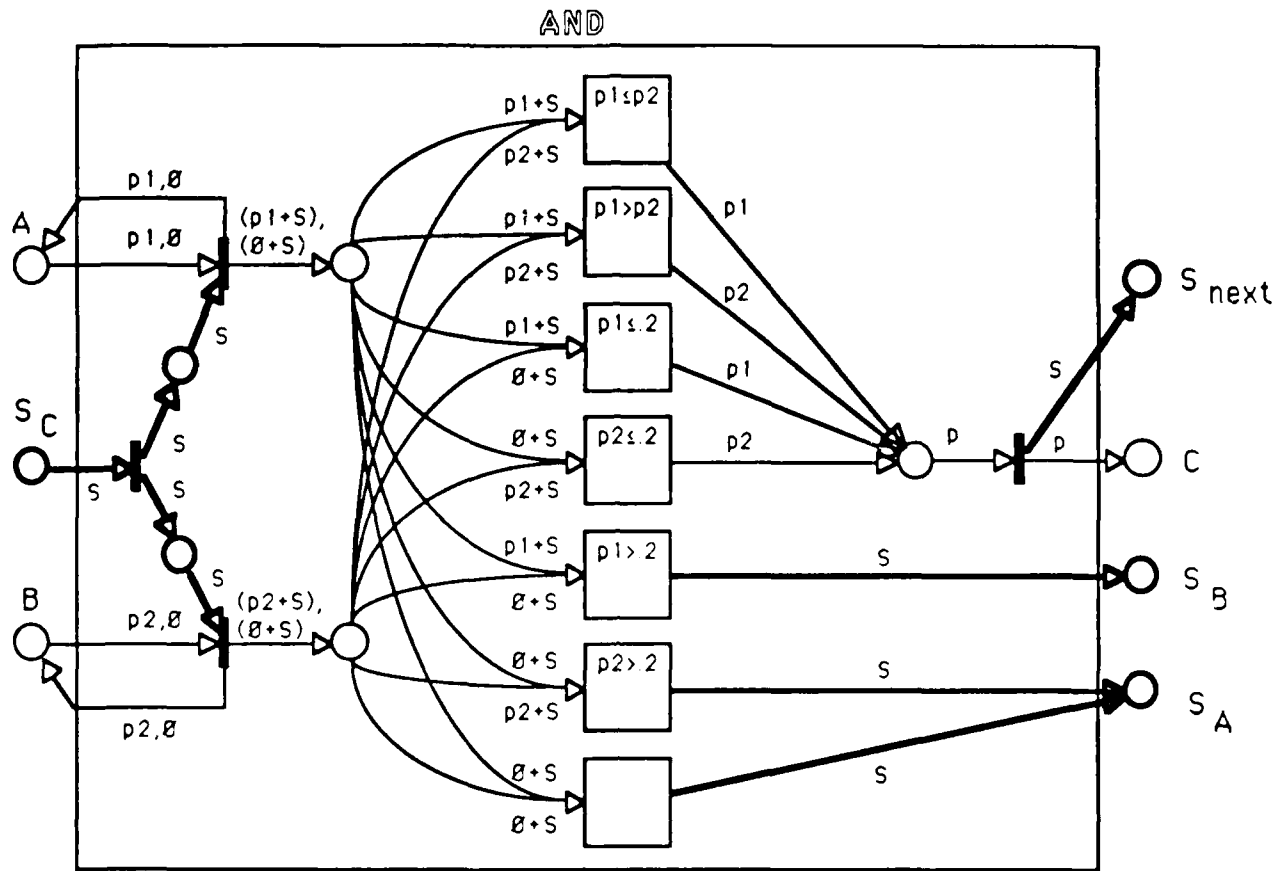
Figure 4.2 Model of the operator AND

The execution of the operation will start only if there is a system token in S_C. We denote by S_C the trigger place of the operator computing C. As soon as there is a token in S_C, the two input transitions are triggered by the allocation of a system token (S) at the input place of these transitions. The values of A and B are therefore reproduced in A and B and in the output place of each of the transitions. These places contain also a system token, which will ensure the enabling of the following transition (i.e., that the two inputs are present). These two places are the input places of seven different transitions which have disjoint conditions of enabling. Only one of these transitions can be enabled and can fire. At the firing, the result, if any, is given in the result place and then in C, while the system token is assigned either to S_{next}, or to S_A, or to S_B according to the result, as previously described.

51
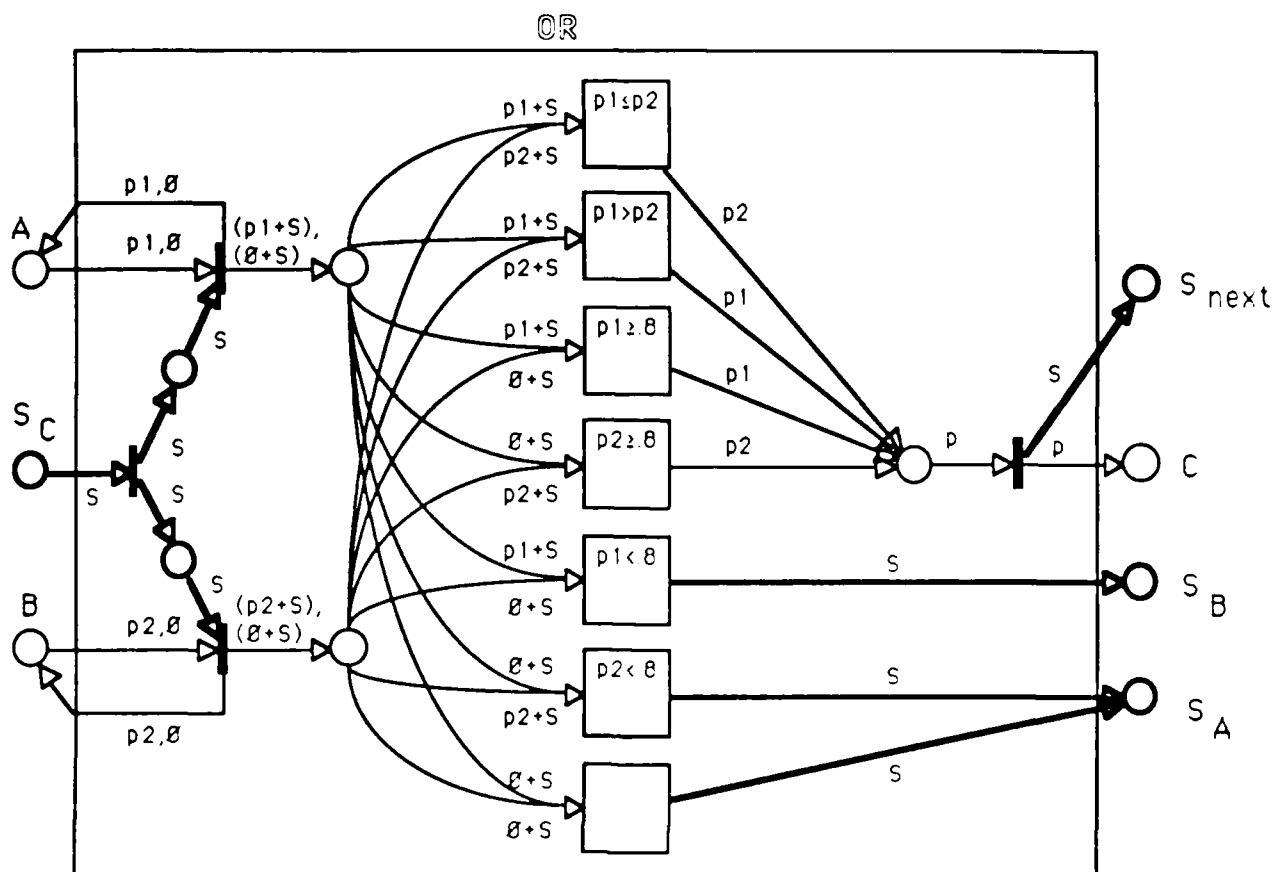
Figure 4.3 Model of the operator OR

These operators can be compounded in super-transitions. The model can be generalized to operators with more than two inputs by combining these basic operators.

An example of the use of these logical operators is shown on the next section, where the representation of a simple inference net is made and the search process in this net is simulated.

NOT
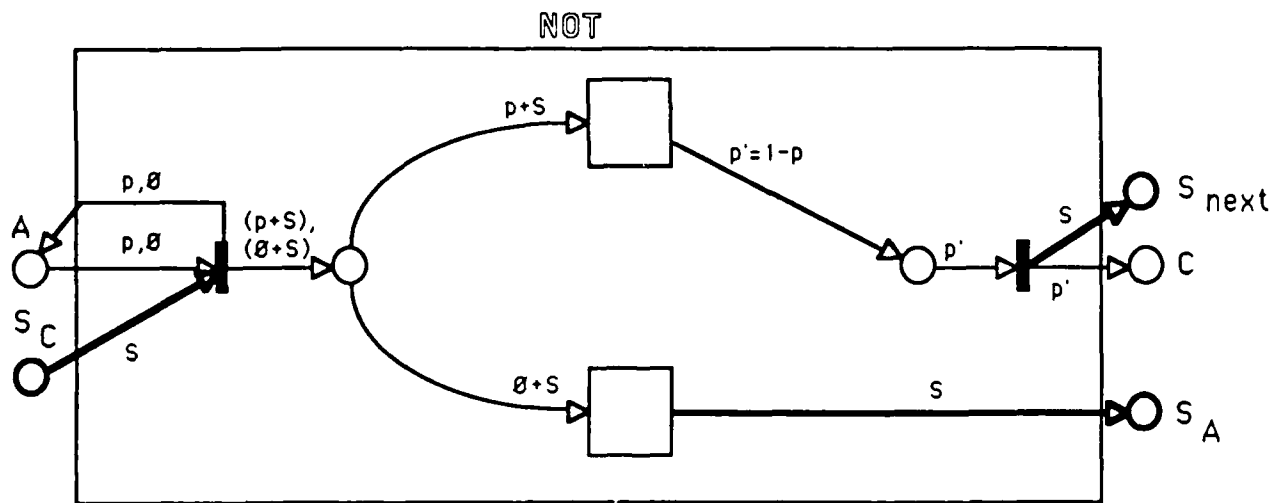
Figure 4.4 Model of the operator NOT

4.4 DYNAMIC REPRESENTATION OF AN INFERENCE NET

The connection of the super-transitions representing the logic operators to places representing the items of evidence leads to a dynamic representation of an inference net. It allows to show explicitly how the inference engine scans the knowledge base. By running a simulation program, we can see in real time what the steps of reasoning are, the possible deadlocks, or mistakes. It allows one to identify the parts of the knowledge base where the knowledge representation is incorrect.

Let us consider the simple symbolic system containing the following rules :

> if A and B => E
> if C and D => D
> if E or F => G

The standard representation of the inference net of this system (see section 3.1) is shown in Figure 4.5.
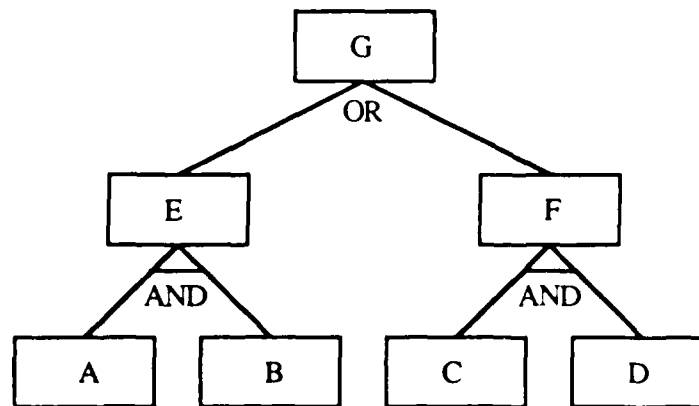
Figure 4.5 Standard representation of the inference net of the example.

The representation of the inference net with Predicate Transition Net is deduced from this representation by :

(1) replacing the rectangles representing the subgoals with the places of our model

(2) replacing the formalism AND, OR, and NOT by the models of the operators aggregated in super-transitions, and linking these places to those transitions (including the self loops).

(3) linking the system places of each operator according to the rules described in section 4.3 for the scheduling of the checking of the unknown subgoals.

The representation of the inference net of the simple symbolic system, using the Predicate Transition Net models of the logic operators is shown on Figure 4.6. The interface module with the user has been added with the places IA, IB, IC and ID, where the user can enter the degrees of truth of A, B, C and D.
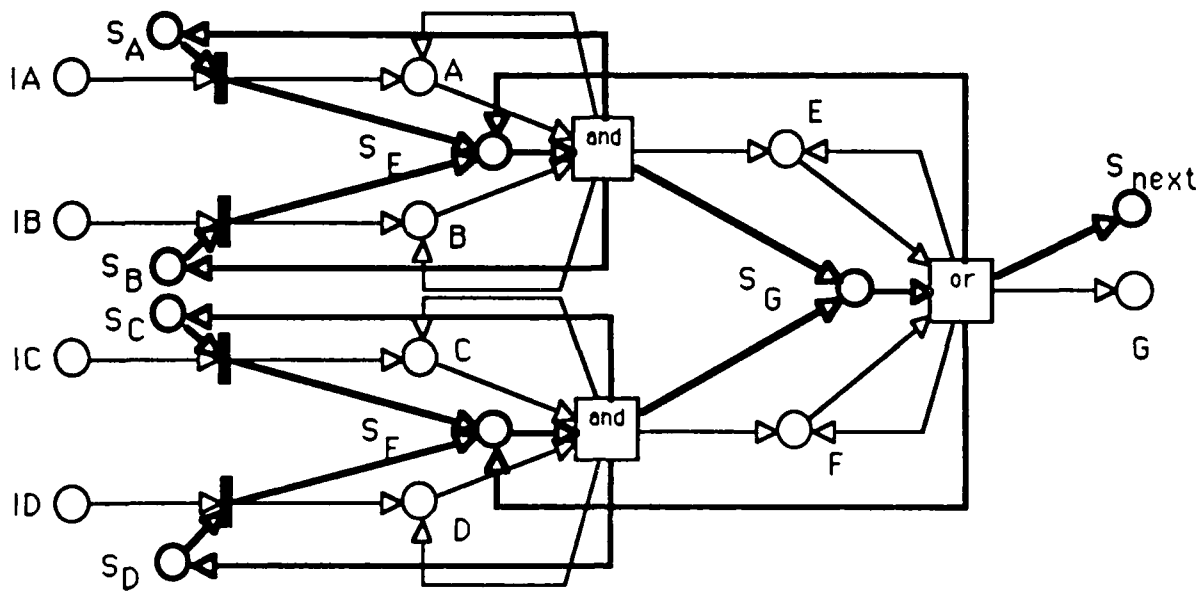
Figure 4.6 Inference net of a simple symbolic system,
using the Predicate Petri Nets formalism

The simulation of the propagation of the tokens in this net allows one to observe the reasoning process followed by the system. The mapping of the different places of the net at each step of the process of the simulation is shown on table 4.1.

The search for the degree of truth of the goal G starts when the system token is put in the system place S_G, at the beginning of the search (step 1). The degree of truth of G cannot be evaluated when the operator OR is executed. The system token is therefore assigned to S_E for the checking of the subgoal E (step 2). The execution of the operator AND cannot lead to a result for E and the system token is allocated to S_A (step3), which triggers an interaction session with the user to get the degree of truth of A. The user enters this value (say 0.9) through IA (step 4) which is assigned to A, while the system token is assigned to S_E (step 5). Since, the degree of truth of A is larger than 0.2, the result of the operator AND cannot be given in E and the system token is assigned to S_B (step 6) to get the degree of truth of B (say 0.8) through IB (step 7). The system token is then re-assigned to S_E to trigger the operator AND (step 8), which can now be executed. The minimum of the degrees of truth of A and B, 0.8, is put in E, while the system token is assigned to S_G (step 9). Since the degree of truth

of E is equal to 0.8, the operation OR can be performed to produce the result G equal to 0.8. The system token is allocated in S_{next} (step 10). The subgoal F has not been checked and all the part of the net which is used to evaluate F has been pruned.

	A	IA	B	IB	C	IC	D	ID	E	F	G	S_A	S_B	S_C	S_D	S_E	S_F	S_G	S_{next}
Step 1																		S	
Step 2																S			
Step 3												S							
Step 4		0.9										S							
Step 5	0.9															S			
Step 6	0.9												S						
Step 7	0.9			0.8									S						
Step 8	0.9		0.8													S			
Step 9	0.9		0.8						0.8									S	
Step 10	0.9		0.8						0.8		0.8								S

Table 4.1 Mapping of the Places at the different steps of the simulation

4.5 APPLICATIONS OF THE MODEL

4.5.1 Assessment of Parallelism

The search for a solution with an expert system is very costly in time and memory storage and some limits exist regarding the size and the kind of problem that can be solved. A way to improve the performance is to dispatch the problem to different processors, each of them solving a part of the problem concurrently. Results of each of the parts are sent to the appropriate processors through a message passing protocol. The problem is that message passing is costly in time. Therefore, the different tasks allocated to each processor have to be chosen very carefully in order to minimize the number of communications among the

processors and also the average idle time in processors waiting for the result of a computation done on another processor. The most reasonable way to do this is to allocate to each processor a part of the problem which is as independent as possible of the other parts of the computation.

The Predicate Transition Net model of the inference net allows to schedule the allocation of the rules to the different processors. This is done in the following way :

We first transform the standard representation of the inference net of the problem to be solved into its representation with Predicate Transition Nets, described in section 4.4. This representation is then modified by first suppressing the system net representing the action of the inference mechanism to select the rules (i.e. all the places and connectors represented in bold style on the Figures),and second, by suppressing the self loops representing the memorizing in the fact base of the intermediate results.

For example, the transformed net obtained from the representation of the example of the simple symbolic system, obtained from its representation shown on Figure 4.6, is shown on Figure 4.7.
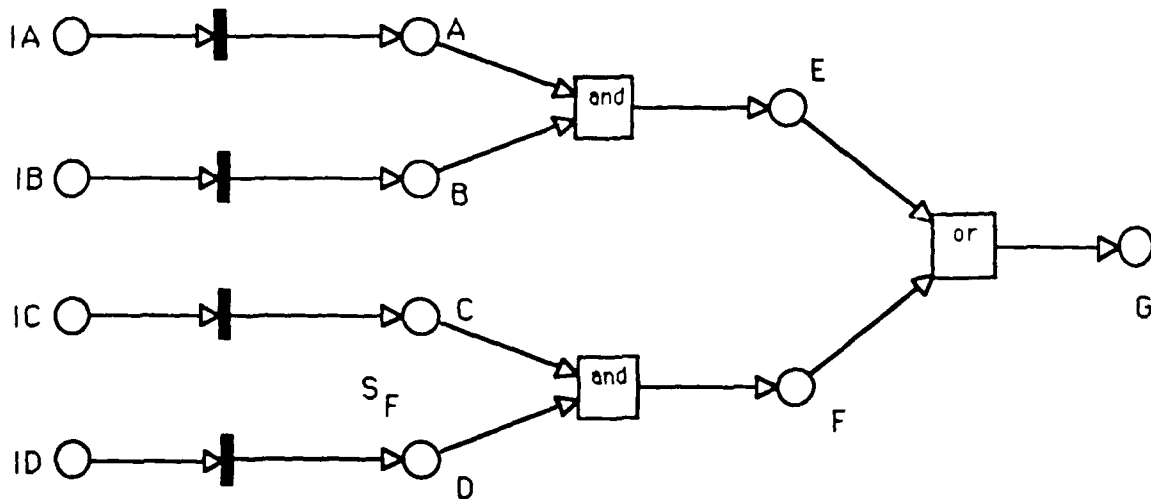
Figure 4.7 Example of transformed net for the assessment of parallelism.

Now, the simple paths and the slices of the transformed net can be determined. The slices indicate the operations which can be performed concurrently, while the simple paths indicate the sets of the dependent rules which have to be scanned sequentially by the system.

For the simple example presented in Figure 4.6, we consider four processors to handle the problem. The slices of the transformed net are :

Slice 1 : { IA, IB, IC, ID}
Slice 2 : {A, B, C, D}
Slice 3 : {E, F}
Slice 4 : {G}

The simple paths are :

Simple path 1 : IA,A,E,G
Simple path 2 : IB,B,E,G
Simple path 3 : IC,C,F,G
Simple path 4 : ID,D,F,G.

An efficient way to allocate the rules to the processors is :

processor 1 : IA,A,E,G
processor 2 : IB,B   (send B to processor 1)
processor 3 : IC,C,F   (send F to processor 1)
processor 4 : ID,D   (send D to processor 3)

This allocation requires only three message passings among the four processors.

4.5.2 Timeliness

The other application of the model is more appropriate to this thesis. The model allows the evaluation of the time needed to produce an ouput ; this is then used to compute the timeliness of an organization using an expert system.

The timeliness of an expert system is related to the number of rules in the rule base scanned by the system to give an answer to a specific problem or goal, and to the number of interactions with the user.

The model we have defined allows a quick identification of the parts of the rule base which have been scanned, given a certain set of inputs, to reach a specific goal, since each place contains the token symbolizing the value of the rule or fact it represents.

Let us consider an expert system being used to give a certain answer in a certain environment. We represent the input X_i to the system as a n-tuplet where n is the total rumber of questions which can be asked by the system. The answer to the questions are contained in this n-tuplet at the location corresponding to the question asked (this may not be listed in the order of appearance in time). The locations for the unasked questions are left empty. We will denote by n_i the number of questions asked by the system. The number of X_i's might be very large but it will be bounded. Given a certain environment, we can define a distribution $p_i(X_i)$ for the appearance of the input X_i.

For a specific input X_i, we can identify N_i, the number of places scanned by the system to reach its goal, since they still contain the degrees of truth of the subgoals they represent. If τ is the average time to check a rule and t is the average time taken by a user to answer a question asked by the system, then the time t_i to get an answer given an input X_i will be :

$$t_i = N_i \tau + n_i t \qquad (4.2)$$

Therefore, the average time of use T of the expert system for the set of inputs X_i will be given by :

$$T = E[t_i] = \sum_i p_i t_i = \sum_i p_i N_i \tau + \sum_i p_i n_i t \qquad (4.3)$$

which leads to :

$$T = E[N_i] \tau + E[n_i] t \qquad (4.4)$$

where E[X] denotes the expected value of the variable X.

The time T obtained is the average time needed to get an answer from the expert system.

In chapter 5, we shall study an organization that uses an expert system. For computing the timeliness, the expert system is represented as a transition with the firing time depending on the input.

CHAPTER 5

AIR DEFENSE COMMAND AND CONTROL

5.1 SCOPE OF THE APPLICATION

An application using the model of the expert system described in chapter 4 has been developed in order to show to what extent the measures of performance of an organization are modified when an expert system is used. In this example, the expert system is used as decision aid for a decisionmaker performing fusion of inconsistent information.

The inconsistency of information can be attributed to different causes. The main cause is the uncertainty of the information, resulting from inaccuracy in measured data or lack of sensor coverage. Presence of noise in data perceived by different decisionmakers, who use different algorithms to filter it, can lead to different perceptions for the same stimulus. Bad interpretation of data can create ambiguities and even conflicts in reports. Finally, in a military context, inconsistency of information can be explained by the attempt by the enemy to mislead the organization about his actions through the use of fakes or jamming technics. This presence of inconsistent information to be fused jeopardizes the correct execution of the mission of an organization.

The role of expert systems in aiding the decisionmaker in performing the information fusion can be therefore critical. The volume of information needed to be collected, sorted and processed to perform the fusion in an efficient manner imposes an overwhelming task to the decisionmaker. Expert systems have a capacity to handle symbolic data which allows the consideration of additional knowledge and facts which are too costly in terms of time, effort, and memory storage to be handled efficiently by the decisionmaker.

The key issue in the use of an expert system is to verify the correctness of information by using additional knowledge. For each instance of contradictory data, it can check if their values are consistent with the knowledge it already has. This checking process is done by the automated generation of hypotheses. Expert Systems provide powerful reasoning capabilities and use a variety of knowledge to combine different kinds of data and infer higher levels of information. This generation process occurs most often just before the actual information

61

fusion which can be performed with a more precise assessment of the situation. In the model of data fusion and support system proposed by Waltz and Buede (1986) for the model of command and control decisionmaking developed by Wohl (1981), this process goes one step further with the automated generation of options (response alternatives) from these hypotheses. These options are proposed to the human decisionmaker for selection and action. Furthermore, in some specific applications, the expert system performs the information fusion itself. An example of such system is HASP / SIAP (see Nii, Fiegenbaum, Anton and Rockmore, 1982) which is a knowledge based system that can identify ships by analysing sonar signatures obtained from different sensors. It solves conflicts of interpretation by predicting the possible future signatures, and matching them with the real ones. In the application of this thesis, the expert system only generates hypotheses, taking into account additional knowledge to give a more precise assessment to the decisionmaker performing the information fusion.

In order to emphasize the interest of using expert system in the information fusion stage, the application compares this strategy to two other strategies which are expected to be used by the decision maker in such context :

(1) the decisionmaker performing the information fusion uses only his own assessment and ignores the assessment of the other decision maker. This strategy is related to the way a human being assigns value to information which is transmitted to him, while executing a specific task. The study of Bushnell, Serfaty and Kleinmann (1987) develops a normative-descriptive approach to quantify the processes of weighting and combining information from distributed sources under uncertainty. Their experimentation has shown that one of the human cognitive biases, which appears in the execution of a task, is the undervaluing of the communications from others, which occurs independently of the quality of the information received. The decisionmaker is, therefore, expected to have the tendency to overestimate his own assessment and to assign a lower value to the assessments the others.

(2) the decisionmaker performs a weighted choice among the contradictory assessments which are transmitted to him and compared to his own. This weighting strategy involves the confidence which can be given to the information and which depends on the manner this information has been obtained, or on its certainty. In many models of organizations facing this problem of inconsistent information and using the weighted choice strategy, measures of

certainty are the basis for the weighting of different items of evidence. Among the methods used, the Bayesian combination described in section 3.2 has given valuable results.

The remainder of this chapter describes in more detail how these strategies have been implemented in the application. The type of organization used is first given, then the mission is described. The details of the implementation are shown with a description of the inputs and of the algorithms used in the different stages of the decisionmaking process. Attention is focused on the information fusion stage where the implementation of each of the strategies is described. Finally, a description of the methods used to obtain the measures of performance is given. The next chapter describes the results obtained and their interpretation.

5.2 THE ORGANIZATION

The organization studied is a hierarchical two-decisionmaker organization with the Petri net representation shown in Figure 5.1.
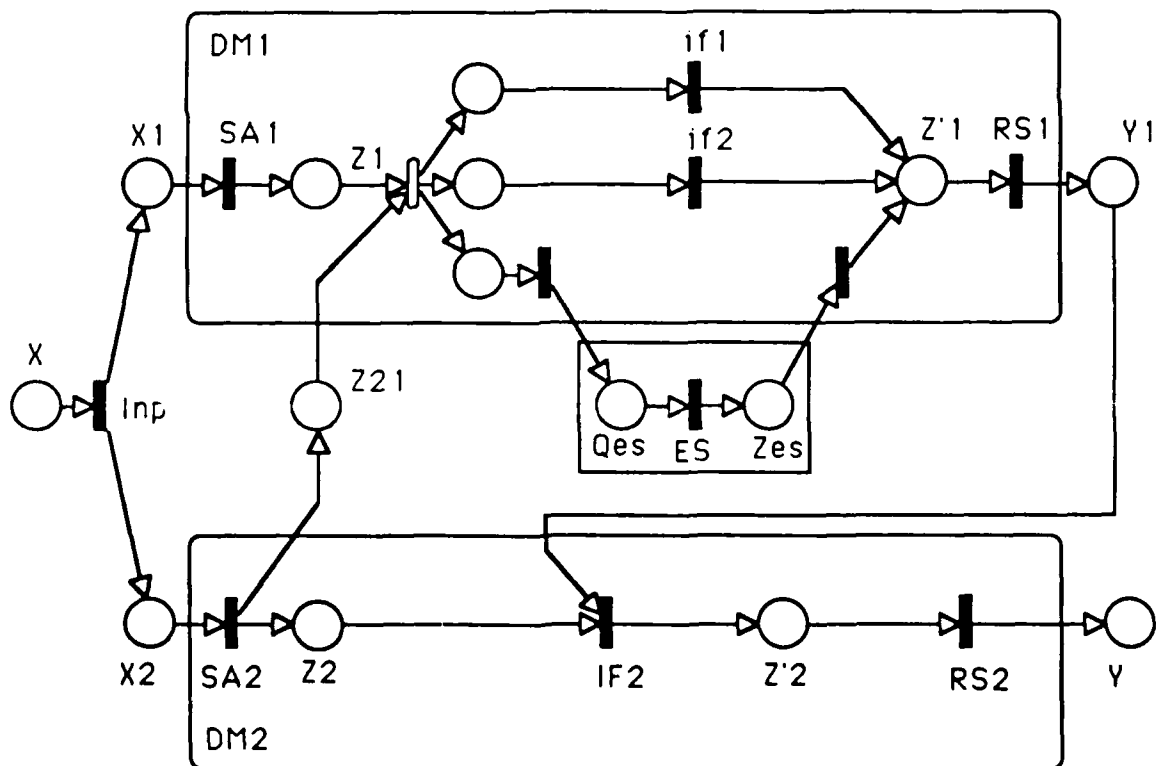
Figure 5.1 Petri net representing the hierarchical 2-DM organization used in the application.

The two decisionmakers, DM1 and DM2, perform their own situation assessment producing the results Z1 and Z2. DM2 sends Z21, which is equal to Z2, to DM1 who is in charge of performing the information fusion with one of the three strategies available. One of them is to use an expert system. With the result Z'1 obtained, the response Y1 is selected and transmitted to DM2. DM2 takes into account this new information in his information fusion stage and realizes the final response selection of the organization, Y.

5.3 CHARACTERISTICS OF THE MISSION

This example is inspired by the game "Ashes" running on Macintosh. In this game, the player has to defend a military base and a population of inhabitants living in three cities against attacking missiles. To destroy these threatening missiles, the player can either send antimissile rockets from the military base (the stock is finite and can be destroyed by missiles), or use a laser beam. When a missile hits a city, the total population decreases. When the military base is hit, the number of antimissile rockets available to the player decreases. The game is over when the population is equal to 0.

In the illustrative application, the organization has to defend a set of facilities composed of three cities, two military bases and two production facilities located in a square with 30 mile sides, as shown on Figure 5.2.

To destroy incoming missiles, the organization can either use a laser beam or send an antimissile rocket. The laser beam is used when the time before the missile hits its target is very short. The antimissile rocket is used in the opposite case. Both weapons require different ways to calculate the meeting point with the missile. The performance of the organization is measured by its ability to send the right weapon at the right place for each incoming missile.
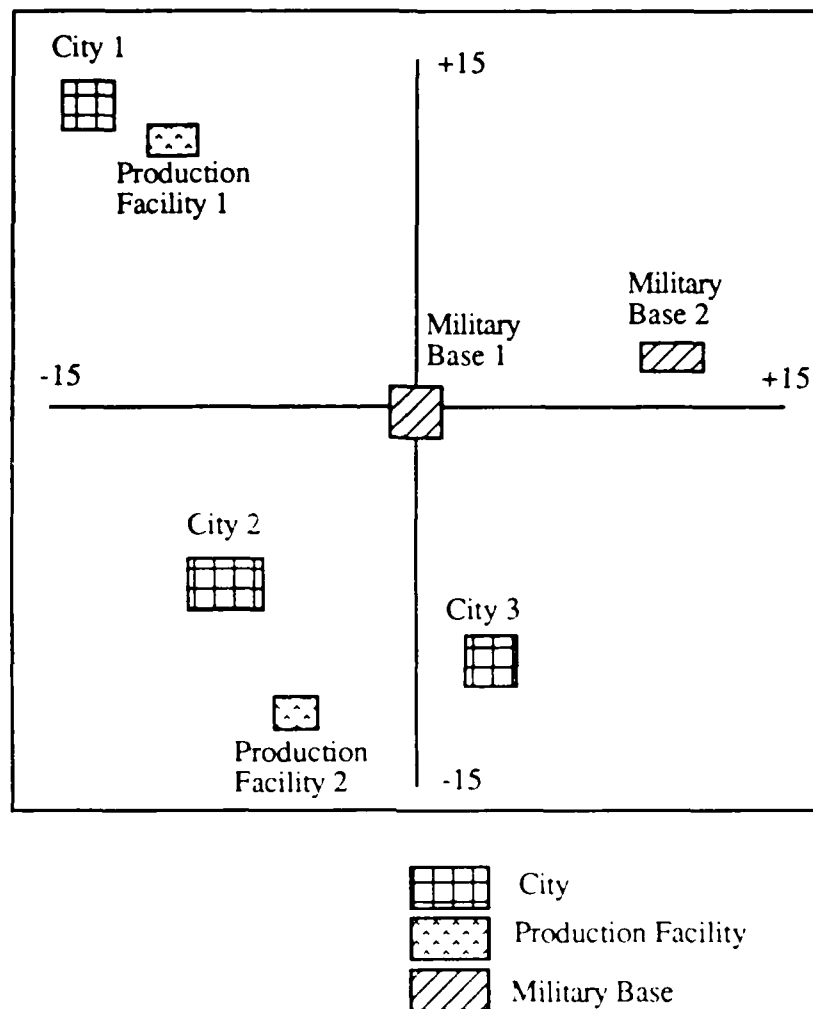
Figure 5.2 Location of facilities to be defended by the organization

5.4 STRUCTURE OF THE INPUTS OF THE ORGANIZATION

We assume that the missiles arrive one at a time and are thus processed by the organization one after another.

Each decisionmaker receives as input two points of the trajectory of the missile. The first one is its position (X,Y,Z) at time t, which is the same for the two decisionmaker to make sure they are assessing the same missile. The second point is the position (Xi,Yi,Zi) [i=1,2] at time $t+1$ as determined by the tracking center of each decisionmaker. The tracking center is

defined as the sum of the human and hardware means assembled to process the information. The use of decoys by the enemy and the presence of noise result in these positions being not the same for each of the decisionmakers. When this is the case, we assume that one of the two is the actual one.

In addition to these different coordinates, the input contains also the confidence factors associated with each position estimate at time t+1. These confidence factors have been generated randomly and measure only the quality that can be attributed to each set of data.

Finally the probability of the input is given, as well as the probability that the right answer will be produced : the type of weapon to be used and the coordinates for sending the weapon to destroy the missile.

An input is therefore a 16-uplet :

(X, Y, Z, X1, Y1, Z1, X2, Y2, Z2, Conf1, Conf2, Prob, weapon, Xtarg, Ytarg, Ztarg)

where :

- X, Y, Z are the longitude, latitude and altitude of the missile at time t.
- X1, Y1, Z1 are the longitude, latitude and altitude of the missile at time t+1 as perceived by the information service of DM1.
- X2, Y2, Z2 are the longitude, latitude and altitude of the missile at time t+1 as perceived by the information service of DM2.
- Conf1 is the confidence factor associated with the data (X1, Y1, Z1).
- Conf2 is the confidence factor associated with the data (X2, Y2, Z2).
- Prob is the probability of the input.
- weapon is the weapon to use. (either beam laser (1) or antimissile rocket (2))
- Xtarg, Ytarg, Ztarg are the longitude, latitude and altitude of the target point.

X's and Y's are integers between -15 and 15.

Z's are integers between 0 and 30. We assume that Z, Z1 and Z2 are greater than 25 and that Z1 and Z2 are less than Z.

There are 33 equiprobable inputs. 24 out of 33 inputs contain inconsistent information. We assume that for half of these inconsistent inputs, the tracking center of DM1 is correct (the tracking center of DM2 is correct for the other half because we assume that for each input, one of the two contradictory positions is right).

The description of the different algorithms used is now given. The flowcharts of the algorithm are given in Appendix A.

5.5 SITUATION ASSESSMENTS OF DM1 AND DM2

DM1 and DM2 have the same situation assessment algorithm. They must perform the two following operations :

(1) The computation of the components of the speed vector of the missile is simply done by subtracting the longitudes, latitudes and altitudes at time $t+1$ and t. DMi produces sx_i, sy_i and sz_i (with $i = 1$ or 2).

(2) The computation of the impact point, i.e. the longitude and the latitude of the missile when its altitude is 0. The operations are :

$$ximp_i = X - Z \frac{sx_i}{sz_i} \tag{5.1}$$

$$yimp_i = Y - Z \frac{sy_i}{sz_i} \tag{5.2}$$

The result produced by DMi is therefore :

$$Zi = (ximp_i, yimp_i, sx_i, sy_i, sz_i) \quad i = 1, 2.$$

DM2 sends Z21 to DM1 with :

$$Z21 = (ximp_2, yimp_2, sx_2, sy_2, sz_2).$$

67

5.6 INFORMATION FUSION OF DM1.

DM1 makes first the comparison between Z1 and Z21. If they are equal, Z'1 = Z1 is produced. If they are different, DM1 has the choice among the three different strategies described in section 5.1.

5.6.1 First Strategy : Ignore Information Sharing.

DM1 produces Z'1 = Z1 without considering the situation assessment Z21, transmitted to him by DM2.

5.6.2. Second Strategy : Weighting of the Confidence Factors.

DM1 considers the confidence factors Conf1 and Conf2 given with the input and which measure the quality of the information to choose Z1 or Z21.

If Conf1 is greater than Conf2, DM1 produces Z'1 = Z1.

If Conf 2 is greater than Conf1, DM1 produces Z'1 = Z21.

If Conf1 is equal to Conf2, DM1 uses the first strategy and produces Z'1 = Z1.

5.6.3 Third Strategy : Use of the Expert System.

DM1 uses an expert system that handles data which would be too costly in terms of effort and workload if they were handled by himself.

The simple knowledge base system which has been developed is described in Appendix B. It evaluates the degree of threat a missile represents as a funtion of the distance between the location of the different facilities and the impact point of the missile estimated by the user. A more sophisticated system could make the assessment of the threat by taking into account the type of missile, the geographical aspect of the area, the direction of winds, the interest for the enemy to destroy the aimed facility, ... This threat assessment is done for the two possible trajectories of the missile, one after another. If the first threat assessment shows that

the target is one of the facilities with enough certainty, the computer stops its search. In the opposite case, the computer evaluates also threat that the missile would have if it followed the second trajectory.

For the evaluation of performance, two types of possible interactions between the decisionmaker and the expert system have been studied :

The user initiated mode when the decisionmaker enters all the data he has in a specified order and the machine gives the result even if all the data are not used in its search process.

The computer initiated mode when the user enters specific data only when the computer asks for them. Only useful data are entered.

The answer of the expert system consists of two numbers between 0 and 1 representing the severity of the threat posed by the missile (according to each assessment). When the answer is given, DM1 does not use a strategy to make a comparison with a result from an internal algorithm, as shown by Weingaertner and Levis (1987). The key point here is that the decisionmaker has not enough data on his own to be able to double check the answer of the decision aid. The decisionmaker produces the final result according to the following rules :

If the degree of threat according to the assessment of DM1 is greater than the one according to the assessment of DM2, the result is $Z'1 = Z1$.

If the degree of threat according to the assessment of DM2 is greater than the one according to the assessment of DM1, the result is $Z'1 = Z21$.

If the two degrees of threat are equal, the problem is not solved and DM1 compares the confidence factors of each set of data in the same way as in the second strategy.

5.7 RESPONSE SELECTION OF DM1

Having chosen the trajectory which seems to be the most likely, DM2 evaluates the threat the missile represents by computing the time remaining before it reaches the impact

point. The computation performed is :

$$t = \frac{\sqrt{(X\text{-}ximpf)^2 + (Y\text{-}yimpf)^2 + Z^2}}{\sqrt{sxf^2 + syf^2 + szf^2}} \qquad (5.3)$$

If this time t is less than a certain threshold, the missile is very threatening (threat = 1) else it is not (threat = 2). DM1 sends to DM2, the type of threat which has just been determined and the fused information : (ximpf, yimpf, sxf, syf, szf).

5.8 INFORMATION FUSION OF DM2.

DM2 executes a simple algorithm which allows him to define the option to follow in the response stage according to the type of threat evaluated and sent to him by DM1.

If the threat is of type 1, the option to follow is option 1.

If the threat is of type 2, the option to follow is option 2.

5.9 RESPONSE SELECTION OF DM2.

This stage determines the response of the organization. DM2 determines the type of weapon to use to destroy the missile and the location to send it.

If the option to follow is option 1, the weapon to use is a laser beam. The location wheie to point the beam is easy to compute and is just the position of the missile on the fused trajectory at time $t + t_{resp}$, where t is the time for which the first set of coordinates (X, Y, Z) was given and t_{resp} is the estimated response time of the organization to carry out its task.

If the option to follow is option 2, the weapon to use is an antimissile rocket which is of course much slower than the laser beam. Complicated computations of intersections of lines in three-dimensional space have not been implemented because they would be done by the targetting computer and, thus, do not add any insight to the application. We have assumed that the speed of the rocket could be adjusted in order to have a travel time almost constant. The location to send the missile is therefore the position of the missile on the fused trajectory

at time $t + t_{resp} + t_{trav}$, where t and t_{resp} are defined as above and t_{trav} is the travel time of the rocket to reach its target.

5.10 MEASURES OF PERFORMANCE

The algorithms we have described above have been implemented on a PC-AT in Turbo Pascal in order to obtain the measures of performance. The model of the Expert System described in Chapter 4 has been implemented and incorporated in this set of algorithms. Measures have been defined for the two types of interaction between the computer and the user. These measures are workload, timeliness and accuracy

5.10.1 Workload

As stated in section 2.1, the workload of a decision maker is computed by adding all the entropies of all the variables used to model the procedures he uses to perform his task. The distributions of all the variables are generated by executing the algorithms for all the inputs described in section 5.4. This process of generation starts with a probability equal to zero for all the values that each variable can take. When the execution of the algorithm is performed with the input X_j having a probability p_j, the internal variable w_i , if it is active, takes the value a_i. The probability mass function of this variable w_1 is updated by adding the probability p_j to the probability this variable had to take the value a_i before the execution of the algorithm with this input X_j. The operation for all the variables w_i affected by the input is:

$$p(w_i = a_i \mid X_1, X_2, ..., X_{j-1}, X_j) = p(w_i = a_i \mid X_1, X_2, ..., X_{j-1}) + p_j \quad (5.4)$$

However, to take into account the effect of the different strategies, the workload of the decisionmakers has to be computed for all the mixed strategies. A mixed strategy is a convex combination of the three pure strategies, and is noted (p_1, p_2, p_3), where p_i, $[i = 1,2,3]$ is the probability of using strategy i in the mixed strategy. p_1, p_2 and p_3 verify :

$$p_1 + p_2 + p_3 = 1 \quad\quad\quad (5.5)$$

To compute the workload of DM1 and DM2 for all the mixed strategies, the system of all the variables has to be divided in three subsystems.

71

The first subsystem is composed of the internal variables of the algorithms for situation assessment of DM1 and DM2. The execution of these algorithms and the values taken by their internal variables for each input do not depend on the strategy chosen in the information fusion stage. Therefore, these algorithms are executed only once for each input to generate the probability mass functions of their internal variables. This subsystem allows to compute the invariant part of the workloads of DM1 and DM2, G_{inv}^1 and G_{inv}^2.

The second subsystem is made of the variables of the different algorithms of the information fusion stage. This subsystem has for input $(Z1,Z21)$ and produces the output Zf with three different algorithms. Each algorithm i is executed independently of the others for all the inputs and the sum of its internal variables (Zf is considered to be an internal variable of each algorithm) gives the activity of coordination of the algorithm of the strategy i, g_c^i. The contribution of this subsystem to the workload of DM1 is evaluated by using the Partition Law of Information given by equation (2.9).

The throughput activity, G_t, is given by :

$$G_t = T(Z1, Z21 : Zf) \qquad (5.6)$$

The blockage term, G_b, is given by :

$$G_b = H(Z1,Z21) - G_t \qquad (5.7)$$

We assume that the data are noiseless and that the algorithm are deterministic. Therefore, the noise, G_n, is only caused by the internal choice in the decisionmaking process and is simply given by :

$$G_n = H(u) \qquad (5.8)$$

where u is the internal choice in the decisionmaking process among the different algorithms. $H(u)$ is equal to :

$$H(u) = H(p_1) + H(p_2) + H(p_3) \qquad (5.9)$$

As stated by Boettcher (1981), the coordination term is given by :

$$G_c = \sum_{i=1}^{3} (p_i g_c^i + \alpha_i H(p_i))$$ (5.10)

where $H(p_i) = - p_i \log_2(p_i) - (1 - p_i) \log_2(1 - p_i)$

and α_i is the number of internal variables of the algorithm i.

We have therefore the activity of the subsystem, $G_{subsystem}$.

$$G_{subsystem} = H(Z1, Z2) + H(u) + \sum_{i=1}^{3} (p_i g_c^i + \alpha_i H(p_i))$$ (5.11)

Finally, since the entropies of Z1 and Z21 have been evaluated in the first subsystem, the contribution $G_{if}(p_1,p_2,p_3)$ of the second subsystem to the workload of DM1 for the mixed strategy (p_1,p_2,p_3) is :

$$G_{if}(p_1, p_2, p_3) = H(u) + \sum_{i=1}^{3} (p_i g_c^i + \alpha_i H(p_i))$$ (5.12)

The third subsystem is composed of the variables of the algorithms used after the information fusion stage. These algorithms are the response selection of DM1, the information fusion and the response selection of DM2. The variables of these algorithms can take three values that are different for each input according to the pure strategy used. Therefore, for each variable of this subsystem, three probability mass functions are generated for all the inputs and for each pure strategy. For each input, the algorithms have to be executed three times, once for each pure strategy. To compute the entropies of these variables for the mixed strategies, a convex probability mass function is deduced from the probability mass functions determined for each pure strategy. By summing these entropies, the variable contribution to the workload of DM1 and DM2 is deduced, $G_{var}^1(p_1,p_2,p_3)$ and $G_{var}^2(p_1,p_2,p_3)$.

73

The workload of DM1 and DM2 can now be evaluated :

$$G_{DM1} = H(input) + H_{inv}^1 + H(Z21) + G_{if}(p_1,p_2,p_3) + G_{var}^1(p_1,p_2,p_3) \qquad (5.13)$$

$$G_{DM2} = H(input) + H_{inv}^2 + H(Y1) + G_{var}^2(p_1,p_2,p_3) \qquad (5.14)$$

5.10.2 Timeliness

The measure of timeliness condidered in this application is related to the response time of the organization. A deterministic processing time has been associated with every algorithm. For the strategy involving the use of the expert system, the time to give an answer has been computed with the method described in section 4.5.2. The implementation of the model of the expert system in the application allows the evaluation of the number of rules scanned by the system for each input to the organization and counts the number of interactions with the user. The response time of the expert system is then deduced. This time is likely to vary with the mode of interactions used.

We assume that DM1 and DM2 perform their situation assessment concurrently and synchronously, and that the same amount of time is needed by the two to give an answer. Therefore, only one of the two processing times is considered. T_{SA1}, (resp. T_{RS1}, T_{SA2}, T_{IF2} and T_{RS2}) denotes the time needed to execute the algorithm for situation assessment of DM1 (resp. response selection of DM1, situation assessment, information fusion, response selection of DM2). $T_{IF1}(i)$ is the time needed to perform the information fusion using the pure strategy i [i = 1, 2, 3]. $T_{IF1}(3)$ is function of the average response time of the expert system computed from its response time for all the inputs. The response time for the strategy i, $T(i)$ is therefore :

$$T(i) = T_{SA1} + T_{IF1}(i) + T_{RS1} + T_{IF2} + T_{RS2}. \qquad (5.15)$$

The response time for each pure strategy (p_1,p_2,p_3) is given by a convex weighting of the response time for each pure strategy. If $T(p_1,p_2,p_3)$ denotes the response time of the organization when the strategy (p_1,p_2,p_3) is used, we have :

$$T(p_1,p_2,p_3) = p_1 T(1) + p_2 T(2) + p_3 T(3) \qquad (5.16)$$

74

5.10.3 Accuracy

Accuracy has been evaluated by comparing the actual response of the organization with the desired or optimal response expected for each input. This desired response is known to the designer. A cost of one has been attributed when the type of weapon to use is not right or when the target point of the weapon to use is not accurate.

For each input X_j having a probability $p(X_j)$, the use of the pure strategy i generates the response Y_{ij} which is compared to the desired response Y_{dj}. The cost function $C(Y_{ij}, Y_{dj})$ has the following characteristics :

$$C(Y_{ij}, Y_{dj}) = 1 \quad \text{if } Y_{ij} \neq Y_{dj}$$
$$C(Y_{ij}, Y_{dj}) = 0 \quad \text{if } Y_{ij} = Y_{dj} \quad (5.17)$$

The accuracy $J(i)$ obtained for the pure strategy i is :

$$J(i) = \sum_{j} p(X_j) C(Y_{ij}, Y_{dj}) \quad (5.18)$$

The accuracy for the mixed strategy (p_1, p_2, p_3), $J(p_1, p_2, p_3)$, is obtained by computing the convex combination of the accuracy for each pure strategy :

$$J(p_1, p_2, p_3) = p_1 J(1) + p_2 J(2) + p_3 J(3) \quad (5.19)$$

The next chapter provides an analysis of the results obtained by using these measures of performance.

CHAPTER 6

RESULTS AND INTERPRETATION

The results of the application introduced in Chapter 5 are described and interpreted in this chapter. The effects of the three strategies on the performance of the organization are first analyzed. Attention is then focused on the effects of the mode of interaction and, finally, the extent to which the handling of uncertainty in the expert system improves the performance of the organization in comparison to an expert system using boolean logic is analyzed.

6.1 MEASURES OF PERFORMANCE

6.1.1 Pure Strategies.

Table 6.1 displays the measures of performance of the organization for each pure strategy as computed by the program.

Table 6.1 Measures of performance of the organization for each pure strategy

	Strategy 1 : Ignoring other assesment	Strategy 2 : Weighted Choice	Strategy 3 : Expert System	Units
J	0.360	0.270	0.210	-
T	18.240	18.960	21.015	seconds
G1	63.414	64.921	70.293	bits per symbol
G2	43.920	43.847	43.240	

The following rankings from the best to the worst can be deduced :

Accuracy : $J(3) > J(2) > J(1)$

Response time : $T(1) < T(2) < T(3)$

Workload :   DM1 : $G_{DM1}(1) < G_{DM1}(2) < G_{DM1}(3)$

                 DM2 :   very small difference among the 3 strategies.

These results confirm the fact that the taking into account of more knowledge, either about the way data are obtained, in the case of the weighted choice strategy, or about the meaning of information, when the expert system is used, yields greater accuracy. Accuracy is an important issue for the kind of mission this type of organization is expected to carry out. The results show also that taking into account more knowledge requires the handling of more data. Therefore, more time is needed and more effort expressed in terms of workload, is *required*. This increase in workload is caused more by the extra decisions which must be made when the knowledge is taken into account than by operations or manipulation done with the additional knowledge. These manipulations are done by the decision aids, out of the control of DM1.

When DM1 ignores the situation assessment of DM2, very few operations are performed. The response time is the smallest of the three. If the measure of timeliness is the ability of the organization to give a response as fast as possible, this strategy leads to a more timely response than the two others. The simplicity of the algorithm results in low workload for DM1 in comparison with the other strategies. This lower workload is explained by the fact that DM1 handles fewer variables. This strategy has low accuracy in comparison with the other strategies, because the choice made on the information to be fused is arbitrary and has no rational justification.

For the weighted choice strategy, no operation on variables received is performed. DM1 makes only a comparison between the weights of the information. We have assumed that the weighting process was carried out outside the organization, by a decision aid or a preprocessor. Therefore, DM1 performs only few operations more than in the first strategy.

Therefore, workload and response time are slightly larger than for the first strategy because of the extra information obtained by comparing the confidence factors. The increase of 3.9 % in response time and of 2.4 % in the workload of DM1 is found. The measure of how the data have been obtained, given through the confidence factors, brings a large gain in accuracy. An improvement of 25 % in the accuracy of the organization in comparison to the first strategy is observed. These results show, as expected, that taking into account the quality of information plays an important role in the accuracy of the organization, without degrading substantially the other measures of performance.

When the expert system is used, the increase in workload of DM1 is about 8.3 % from the level of strategy 2, and 10.8 % from the level of the first strategy. This can be explained by the handling by DM1 of the assessments given by the expert system. These assessments are variables which have greater entropies and which require also more processing. The increase in response time (of 10.8 % from the level of strategy 2 and of 15.2 % from the level of strategy 1) is mainly caused by the time taken by DM1 to interact with the system and the time needed to get the answer. This response time of the expert system can get larger as the size of the knowledge base and of the magnitude of the problem to solve increase. In the example, the simplicity of the system hides the real effect on timeliness which can be expected with the use of such interacting system. The gain in accuracy is very significant, about 22 %, in comparison with the accuracy reached with the second strategy and 41.7 % from the level reached when the situation assessment of the other is ignored. This shows the extent to which the accuracy is improved when additional knowledge is used to verify the correctness of information. By using the expert system to evaluate the target of the missile and to estimate the severity of the threat for each possible trajectory, DM1 has a broader assessment which allows him to perform more accurate information fusion, as shown by the results.

Finally, we note that the workload of DM2 remains almost constant for all the strategies. A variation of 1.5 % can be observed. He uses always the same algorithms, and only the different distributions of the variables of the algorithms obtained, when different strategies are used by DM1, explain this small variation in his workload.

6.1.2 Mixed Strategies

The performance measures (accuracy, timeliness, and workload of DM1) obtained by the organization, when mixed strategies are used by DM1 in his information fusion stage, are represented by a surface in the space (G1, T, J). The projections of this surface on the planes (J-G1), (J-T) and (T-G1) are drawn on Figures 6.1, 6.2, and 6.3.
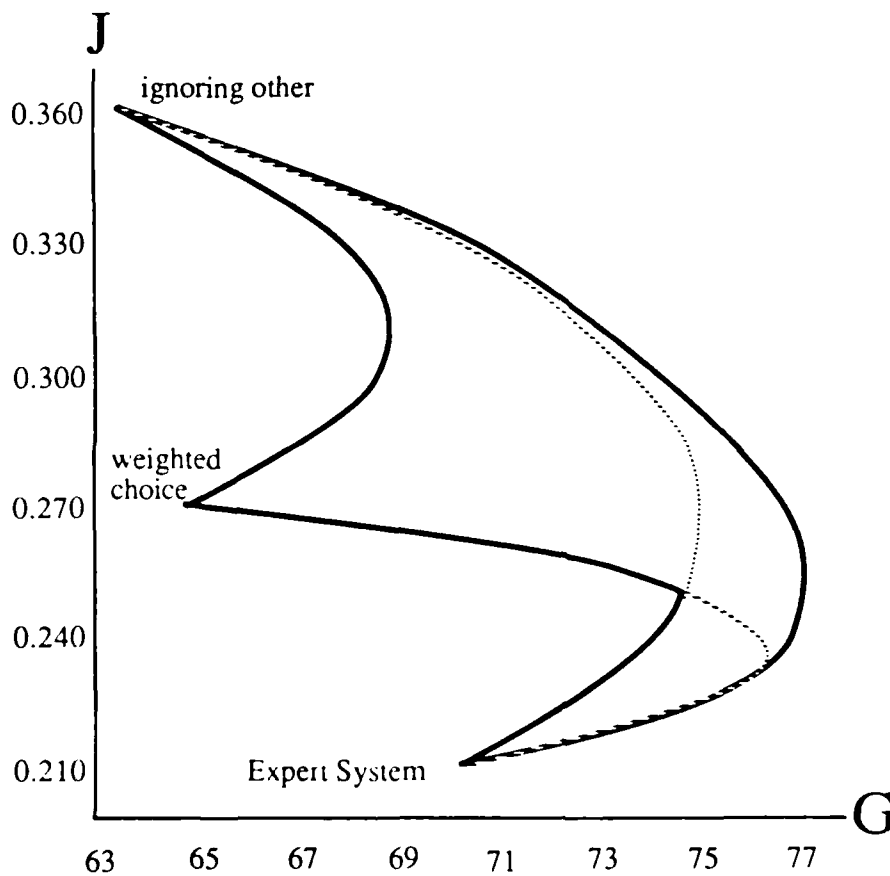
Figure 6.1 Mixed Strategies : performance and workload for DM1

On Figure 6.1, accuracy and workload reached for each pure strategy are located at the three cusps of the figure. The convex combination of two pure strategies only gives the expected U-shaped curve which is explained by the binary entropy terms in equation (5.10), (5.11), and (5.12).
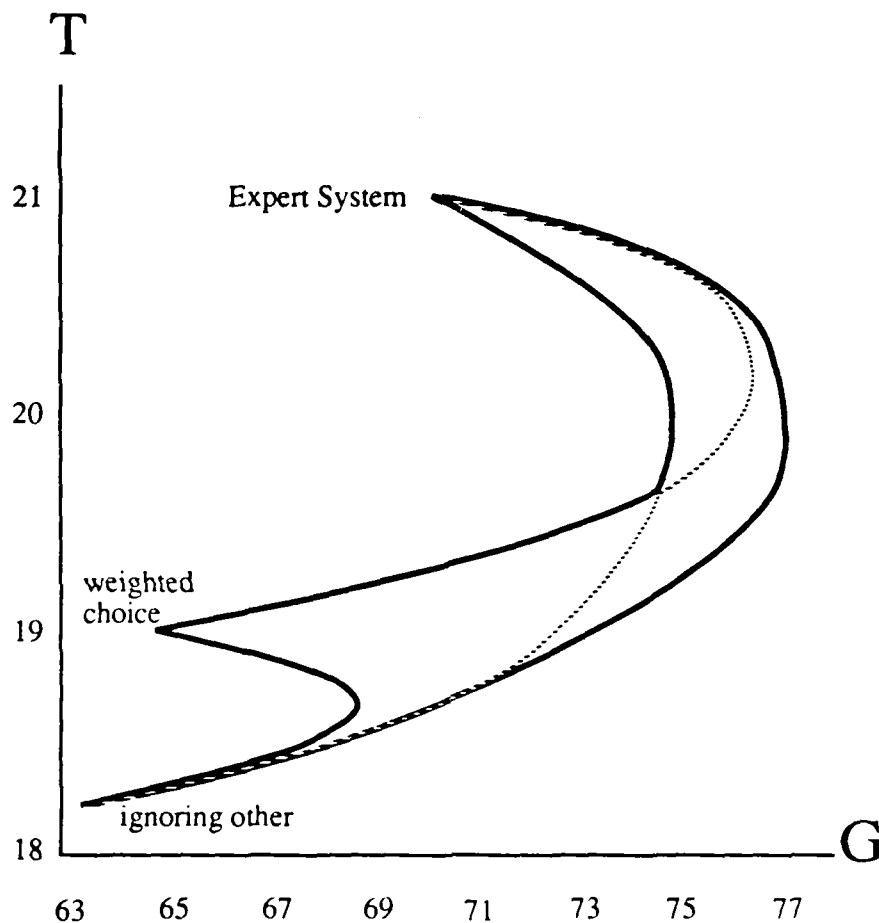
Figure 6.2 Mixed Strategies : Timeliness and Workload for DM1

The projection of the locus on the plane T-G shows the correlation between timeliness and the workload of DM1. The same kind of U-shaped curves for the convex combination of two pure strategies only are observed.
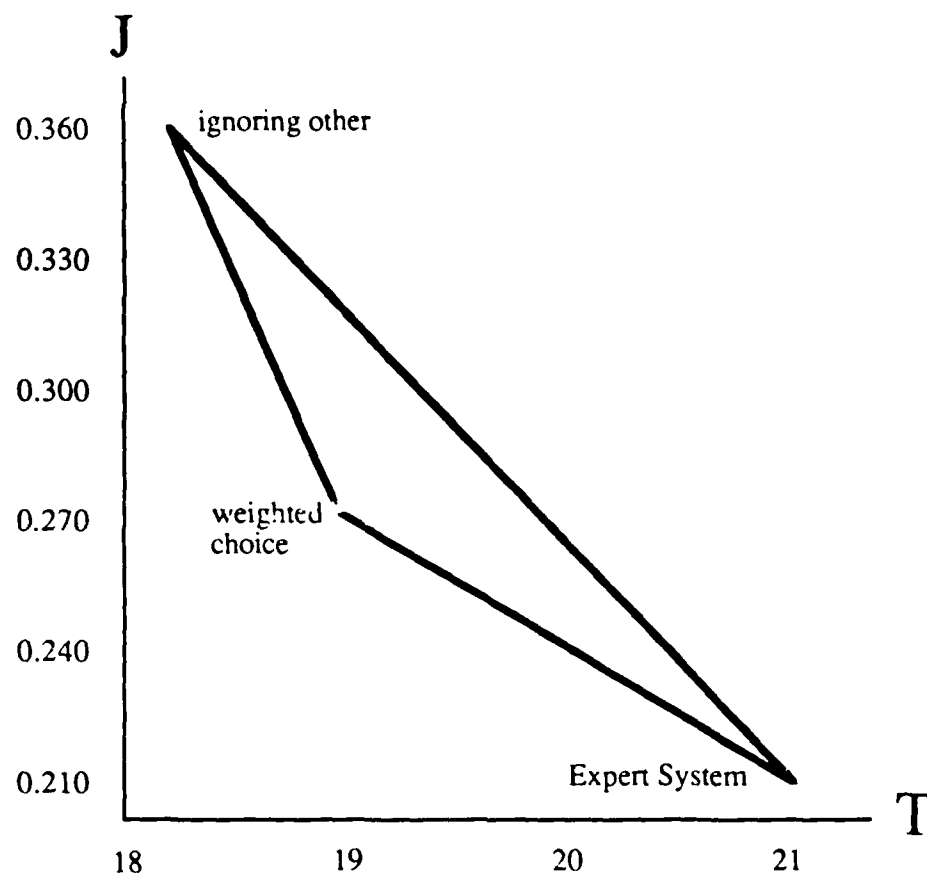
Figure 6.3 Mixed Strategies : Accuracy and Timeliness of the Organization

Figure 6.3 shows that, for two pure strategies only, J and T are linear combinations of each other. Therefore, the projection of the locus J - T - G1 on the plane J - T is a triangle of which the corners are the level reached in accuracy and response time for each pure strategy.

Figure 6.3 shows also that the accuracy obtained by using the expert system is at the cost of longer response time. Therefore, in the case of a mission in which the constraints in time are too binding, the expert system is useless because in this case, too much time would be needed to perform the information fusion.

6.2 EFFECT OF THE MODE OF INTERACTION

The two modes of interaction have been implemented in the program and the results contained in Table 6.2 are obtained.

Table 6.2 : Effect of the mode of interaction on the Measures of Performance

	Strategy 1 : Ignoring other assesment	Strategy 2 : Weighted Choice	Strategy 3 : Expert System	
			user initiated	computer initiated
J	0.360	0.270	0.210	0.210
T	18.240	18.960	21.015	20.850
G1	63.414	64.921	70.293	70.293
G2	43.920	43.847	43.240	43.240

There is no change in accuracy or workload, however, a slight change in timeliness is observed. This is caused by the fact that in the user initiated mode of interaction, all the data which have a chance to be processed by the expert system are entered at the beginning of the session. Even the data which are useless for the resolution of the specific problem are inputs of the system. In the application, the position of the impact points according to the two different situation assessments are entered, even if the first set is sufficient to assess the degree of threat of the missile. Therefore, more time is needed than in the computer initiated mode where data are entered at the request of the system during the search. In the application, DM1 enters first the coordinates of the impact point he had computed in the situation assessment stage. If the expert system asseses that the degree of threat of the missile is below a certain level, it asks to DM1 to enter the second set of coordinates. In this way, only the useful portion of the set of data is processed leading to a reduction in the response time.

It is important to note that in the application no workload have been assigned to the process of entering the information to the expert system. The process in the example is only a reproduction of the information the decisionmaker has. If the inputs asked by the expert

system do not correspond to the data the decisionmaker has, he would have to perform small algorithms to deduce these inputs from the information he has. Let us consider an example where the decisionmaker has computed or received from another member of the organization the value of the speed of an object being analyzed. If the expert system asks the decisionmaker the question : "speed of the object : [possible answer : low, moderate, high]," the decision maker will have to deduce from the actual value of the speed the attribute asked by the system. A small algorithm will have to be executed, increasing its workload. It can be expected therefore that, in this case, a change in workload similar to the change in response time would be observed. This issue raises the problem of the adequate design of the expert system, or more generally, of the decision aid in which the mode of interaction has to be thought very carefully to avoid an unnecessary increase in the workload of the decisionmaker and in the response time.

6.3 FUZZY LOGIC VS. BOOLEAN LOGIC

For this illustrative application, the levels of performance reached when different expert systems are used, have been studied. The performance achieved with an expert system using fuzzy logic as the means of inference, which has been developed for the application, has been compared to the performance obtained by using an expert system which does not deal with uncertainty and uses classic boolean logic. This version of the expert system has been obtained by changing the mapping functions (only values 0 or 1 could be processed instead of the real numbers between 0 and 1). While the assessments of the threat of the missile for each trajectory given by the expert system using fuzzy logic are two numbers between 0 and 1, the assessments given by the expert system using boolean logic have only the values true or false. The different measures of performance obtained for the two systems are summarized in the Table 6.3.

The following conclusions can be drawn from these results :

The organization has a response time slightly lower with an expert system using boolean logic than with the expert system using fuzzy logic (2.3 %). This is due to the fact that by assigning the value true or false to the severity of threat, the system can reach a conclusion (which is not always the best one) by examining fewer possibilities. It can prune a larger part of the knowledge base than the fuzzy logic system when it reaches the conclusion that the

missile is threatening a specific facility. When this conclusion is reached for the first possible trajectory, the other trajectory is not examined. This results in a shorter time to produce the answer and in fewer interactions with the user. Therefore, a shorter response time of the organization is observed.

Table 6.3 : Comparison of measures of performance
for an expert system using fuzzy logic and an expert system using boolean logic.

	Expert System - Fuzzy Logic		Expert System - Boolean Logic	
	user initiated	computer initiated	user initiated	computer initiated
J	0.210	0.210	0.240	0.240
T	21.015	20.850	20.974	20.364
G1	70.293	70.293	65.969	65.969
G2	43.240	43.240	43.287	43.287

Since the expert system with boolean logic assesses the threat only with the value true or false, the answer of the expert system has a lower entropy. The workload of the decisionmaker is therefore lower (about 6.8 %) when he uses the expert system with boolean logic than when he uses the expert system with fuzzy logic.

By pruning a larger part of the knowledge base when it reaches a conclusion, the system has more chance to make the wrong assessment of the threat. The results show that, indeed, the system with boolean logic exhibits lower accuracy than the system with fuzzy logic. The level of accuracy is, nevertheless, better than for the two other strategies expected to be used in the information fusion stage and is explained by the fact that more knowledge is taken into account in the information fusion process.

These results emphasize the role of fuzzy logic for this kind of problem taking into account magnitudes which are a matter of degree. The use of boolean logic for this kind of

problems implies the definition of thresholds for these magnitudes. This lack of flexibility in the reasoning appears to yield lower performance than with fuzzy logic.

CHAPTER 7

CONCLUSION AND DIRECTIONS FOR FUTURE RESEARCH

7.1 CONCLUSION

In this thesis, a model of an expert system with fuzzy logic as a means for dealing with uncertainty has been developed using the Predicate Transition Nets formalism. This has been done through the modeling of the basic logic operators AND, OR and NOT. The combination of these operators makes it possible to represent the inference net of a consultant expert system using production rules and to study its behavior dynamically. A method to make time-related measures from this representation has been introduced, taking into account the portion of the rule base scanned by the system and the number of interactions with the user.

This model of an expert system has then been applied to an example of a two decisionmaker organization facing the problem of fusion of inconsistent information. The decisionmaker must identify the trajectories of missiles that they then have to destroy to protect a set of facilities. In this application, the expert system helps the decisionmaker to clarify the contradictory situation assessment he has to fuse. This strategy has been compared to two others expected to be used in this situation : (1) ignoring the assessment of the other decisionmaker, (2) making a weighted choice among the two contradictory situation assessments, by taking into consideration the way the data used to produce these assessments have been obtained by each decisionmaker. Measures of performance (workload, timeliness and accuracy) have been evaluated. The results show that the use of the expert system improves significantly the accuracy of the organization, but requires more time and increases the workload of the decisionmaker using it. The comparison of the two modes of interaction between the user and the system has shown variations in workload and in response time : the computer initiated mode requires less workload and less response time for a same level of accuracy. This result tends to show that the design of an interacting decision aid must take into account not only the characteristics of the problem to be solved, but also the way the decisionmaker would use it.

7.2 DIRECTIONS FOR FUTURE RESEARCH

Several directions for future research seem promising for studying the use of expert systems in decisionmaking organizations.

One direction is the extension of the model of expert systems developed in the first part of the thesis. This model does not allow the representation of rules, each consequence of which only brings a partial certainty in the conclusion, similar to the certainty factor approach described in section 3.2. Such rules are of the kind :

IF   the object has delta wings
THEN the object is a fighter plane with a certainty of 0.8

IF   the object flies at great speed
THEN the object is a fighter plane with a certainty of 0.6

This standard inference net representation of this kind of rules is shown on Figure 7.1.
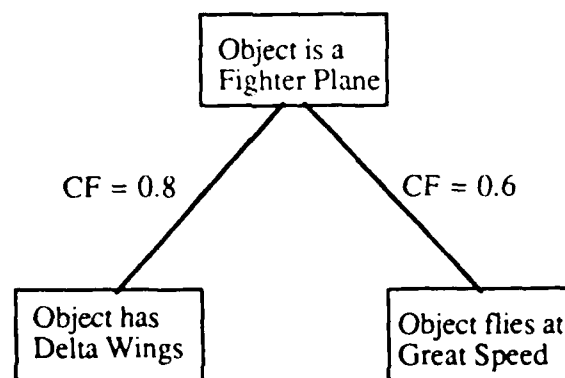
Figure 7.1 : Uncertain Combination of Evidence
in the Standard Inference Net formalism

The combination of these different items of evidence tends to reinforce the belief in the conclusion.

Let us consider the rules :

IF A THEN C with certainty CF_A

IF B THEN C with certainty CF_B

where CF_A (resp. CF_B) is a real number between 0 and 1 representing the certainty of the conclusion C given the hypothesis A (resp. B). The degree of truth of C, $dt(C)$ is determined from the degrees of truth of A and B, $dt(A)$ and $dt(B)$, by the relation :

$$dt(C) = CF_A \, dt(A) + (1 - CF_A \, dt(A)) \, CF_B \, dt(B) \qquad (7.1)$$

which can be written :

$$dt(C) = CF_A \, dt(A) + CF_B \, dt(B) - CF_A \, CF_B \, dt(A) \, dt(B) \qquad (7.2)$$

The extension is therefore to model an operator using Predicate Transition Nets which would handle two kinds of tokens representing the degrees of truth and the certainty factors and which would perform the operation described in equation (7.2).

The second direction proposed is the modeling of an expert system which will be used in the response selection stage and which will assist in the planning of the response by taking in consideration not only the parameters of the problem, but also the resources available to solve it. This model of the planning process for the selection of the response would utilize a set of rules which can be gathered in a knowledge base and would generate, in a consistent manner, the appropriate response. The performance of an organization using this planning system could be evaluated in a similar way to the one developed in this thesis. By considering not only the generation of hypotheses (Waltz and Buede, 1986) deduced from the contradictory sets of data provided by the user, but also the construction of response alternatives, the use of such a model could be an interesting extension of the application described in this thesis.

Finally, the definition of a general methodology for designing decision aids could be an interesting extension of the study of decisionmaking organizations. In the application, the measures of performance of the organization have shown to be very sensitive to the mode of

interaction, the kind of decision aid used (boolean logic system vs. fuzzy logic systems) and the definition of the mapping functions deducing the degree of truth of the pieces of evidence handled by the expert system from the data entered by the user. Therefore, the methodology used to study decisionmaking organizations can be extended for the design of interacting decision aids. For a given set of inputs, design alternatives could be compared and validated to select the one which gives the best measures of performance.

REFERENCES

Andreadakis, S. K. and Levis, A. H., 1987, "Accuracy and Timeliness in Decisionmaking Organizations," *Proc. 10th IFAC World Congress*, Pergamon Press, New York.

Bejjani, G. J and Levis, A. H., 1985, "Information Storage and Access in Decisionmaking Organizations," *Proc. 8th MIT/ONR Workshop on C³ Systems*, LIDS-R-1519, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Boettcher, K. L., 1981, *An Information Theoretic Model of the Decisionmaker*, S.M. Thesis, LIDS-TH-1096, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Boettcher, K. L. and Levis, A. H.,1982a, "Modeling the Interacting Decisionmaker with Bounded Rationality," *IEEE Transactions on Systems, Man and Cybernetics*, SMC-12, No 3, pp. 334-344.

Boettcher, K. L. and Levis, A. H., 1982b, "On Modeling Teams of Interacting Decisionmakers with Bounded Rationality," *Automatica*, Vol. 19, No.6 (1983), pp. 703-709.

Buchanan, B. G. and Shortliffe, E. H., 1984, *Rule Based Expert Systems*, Addison Wesley Publishing Company, Reading, MA.

Bushnell, L. G., Serfaty, D. and Kleinmann, D. L., 1987, *Human Information Processing in Distributed Command and Control*, Report, Cyberlab, University of Connecticut and Alphatec, Inc.

Chyen, G. H.-L and Levis, A. H., 1985, "Analysis of Preprocessors and Decision Aids in Organizations," *Proc. IFAC/IFIP/IFORS/IEA Conference on Analysis, Design and Evaluation of Man-Machine Systems*, Varese, Italy.

Conant, R. C., 1976, "Laws of Information Which Govern Systems," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-6, No.1, pp. 240-255.

Duda, R. O., Hart, P. E. and Nilsson N. J., 1985, "Subjective Bayesian Methods for Rule-Based Inference Systems," in *Readings in AI*, eds. Webber, B. L. and Nilsson, N. J., Morgan Kaufmann Publishers, Inc., Los Altos, CA.

Forsyth, R. , 1984, *Expert Systems, Principles and Case Studies*, Chapman and Hall, London, New York.

Genrich H.J., Lautenbach, K., 1981, "System Modeling with High Level Petri Nets", *Theoretical Computer Science, 13* , pp. 109-136.

Hall, S. A. and Levis, A. H., 1984, "Information Theoretic Models of Memory in Human Decisionmaking Models", *Proc. 9th IFAC World Congress*, Pergamon Press, New York.

Hillion, H., 1986, *Performance Analysis of Decisionmaking Organizations Using Timed Petri Nets,* S.M. Thesis, LIDS-TH-1, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Jensen, K., 1981, "Coloured Petri Nets and The Invariant Method," *Theoretical Computer Science, 14* , pp. 317-336.

Jin, V. Y., Levis, A. H. and Remy, P., 1986, "Delays in Acyclical Distributed Decisionmaking Organizations," *Proc. IFAC Symposium on Large Scale Systems : Theory and Applications,* Zurich, Switzerland.

Johnson, L. and Keravnov, E. T., 1985, *Expert Systems Technology, A Guide,* Abacus Press, Tunbridge Wells.

Jones, P. and Morton, P., 1982, "REVEAL" in *Computer Based Planning Systems,* eds. T. Naylor and M. Mann, Oxford, Ohio : Planning Executive Institute.

Levis, A. H., 1984, "Information Processing and Decisionmaking Organizations : A Mathematical Description," *Large Scale Systems* , Vol. 7, pp. 151-163.

Nii, P., Fiegenbaum, E., Anton, J. and Rockmore, A., 1982, "Signal-to-Symbol Transformation : HASP/SIAP Case Study," *The AI Magazine,* Vol. 3, No. 2, pp. 23-35.

Peterson, J. L., 1980, *Petri Net Theory and the Modeling of Systems* , Prentice Hall, Inc., Englewood Cliffs, NJ.

Reisig, W., 1985, *Petri Nets, An Introduction,* Springer Verlag, Berlin.

Remy P., Levis A. H., and Jin V. Y. Y., 1987, "On the Design of Distributed Organizational Structures," *Proc. 10th IFAC World Congress,* Pergamon Press, New York.

Shannon, C.E. and Weaver W., 1949, *The mathematical theory of communication,* University of Illinois Press, Urbana, IL.

Tabak, D. and Levis, A. H., 1984, "Petri Net Representation of Decisions Models," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-15, No.6, pp. 812-818.

Waltz, E. L. and Buede, D. M., 1986, "Data Fusion and Decision Support for Command and Control," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-16, No.6, pp. 865-879.

Weingaertner, S.T. and Levis A. H., 1987, " Evaluation of Decision Aiding in Submarine Emergency Decisionmaking," *Proc. 1987 Symposium on C^2 Research,* National Defense University, Fort Mc Nair, Washington DC.

Whalen, T. and Schott, B., 1983, "Issues in Fuzzy Production Systems," *International Journal of Man-Machine Studies,* Vol. 19, pp. 57-71.

Wohl, J. G., 1981, "Force Management Decision Requirements for Air Force Tactical Command and Control," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-11, No.9. pp. 618-639.

Zadeh, L. A., 1965, "Fuzzy Sets," *Information and Control* , Vol. 8, pp. 338-353.

Zadeh, L. A., 1983, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *Fuzzy Sets and Systems*, Vol. 11, pp. 199-227.
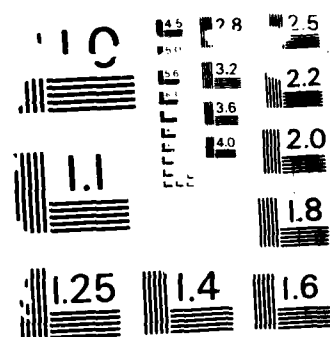
RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963

APPENDIX A

FLOWCHARTS OF THE ALGORITHMS

A.1 SITUATION ASSESSMENT OF DM1

Inputs :

X, Y, Z : coordinates (latitude, longitude and altitude) of the missile at time t

X1, Y1, Z1 : coordinates of the missile at time t+1, according to the information service of DM1.

```
┌─────────────────────────┐
│   X,Y,Z,X1,Y1,Z1        │
└─────────────────────────┘
             │
             ▽
┌──────────────────────────────┐
│ ┌──────────────────────────┐ │
│ │      sx1=X1-X            │ │
│ │      sy1=Y1-Y            │ │
│ │      sz1=Z1-Z            │ │
│ └──────────────────────────┘ │
│             │                │
│             ▽                │
│ ┌──────────────────────────┐ │
│ │ ximp1=X-(Z*sx1/sz1)      │ │
│ │ yimp1=Y-(Z*sy1/sz1)      │ │
│ │      zimp1=0             │ │
│ └──────────────────────────┘ │
└──────────────────────────────┘
             │
             ▽
┌─────────────────────────┐
│  Z1=(sx1,sy1,sz1,       │
│     ximp1,yimp1)        │
└─────────────────────────┘
```
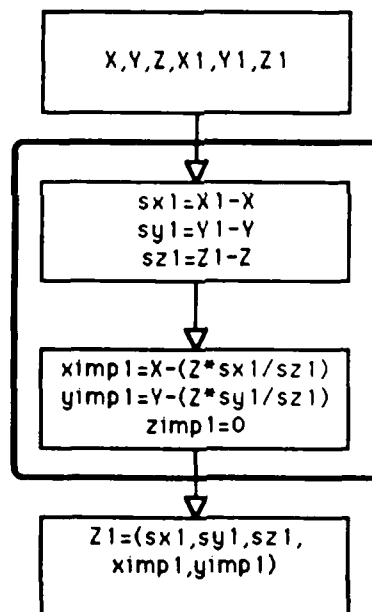
Output :

Z_1 : result of the situation assessment of DM1

sx1,sy1, sz1 : velocity in x, y and z according to DM1.

ximp1, yimp1 : latitude and longitude of the impact point of the missile according to DM1.

A.2 SITUATION ASSESSMENT OF DM2.

Inputs :

    X, Y, Z : coordinates of the missile at time t

    X2, Y2, Z2 : coordinates of the missile at time t+1, according to the information service of DM2.

```
┌─────────────────────────────┐
│      X,Y,Z,X2,Y2,Z2         │
└─────────────────────────────┘
┌─────────────────────────────────┐
│   ┌─────────────────────────┐   │
│   │      sx2=X2-X           │   │
│   │      sy2=Y2-Y           │   │
│   │      sz2=Z2-Z           │   │
│   └─────────────────────────┘   │
│              ↓                   │
│   ┌─────────────────────────┐   │
│   │  ximp2=X-(Z*sx2/sz2)    │   │
│   │  yimp2=Y-(Z*sy2/sz2)    │   │
│   │      zimp2=0            │   │
│   └─────────────────────────┘   │
└─────────────────────────────────┘
┌─────────────────────────────┐
│  Z2=Z21=(sx2,sy2,sz2,       │
│      ximp2,yimp2)           │
└─────────────────────────────┘
```
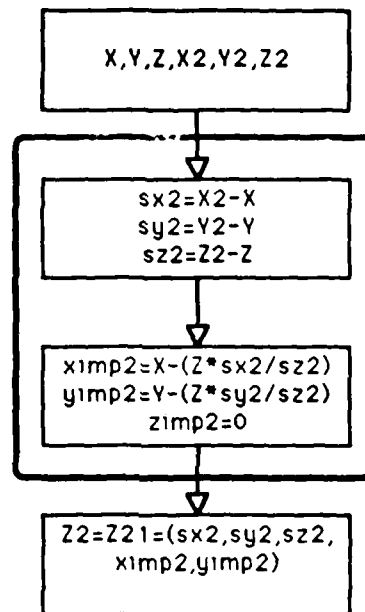
Output :

    Z_2 : result of the situation assessment of DM2,

    Z_{21} : information transmitted to DM1,

    sx2,sy2, sz2 : velocity in x, y and z according to DM2,

    ximp2, yimp2: latitude and longitude of the impact point of the missile according to DM2.

A.3 INFORMATION FUSION of DM1.

Inputs :

Z_1 and Z_{21} : results of the situation assessments of DM1 and DM2

rel1, rel2 : confidence factors of the data given to DM1 and DM2 (part of the input)

Variables :

as1, as2 : assessments of the expert system : severity of threat by the missile according to the trajectories assessed by DM1 and DM2.

Output :

Zf : fused information.

sxf, syf, szf : fused velocities

ximpf, yimpf : fused coordinates of the impact point.
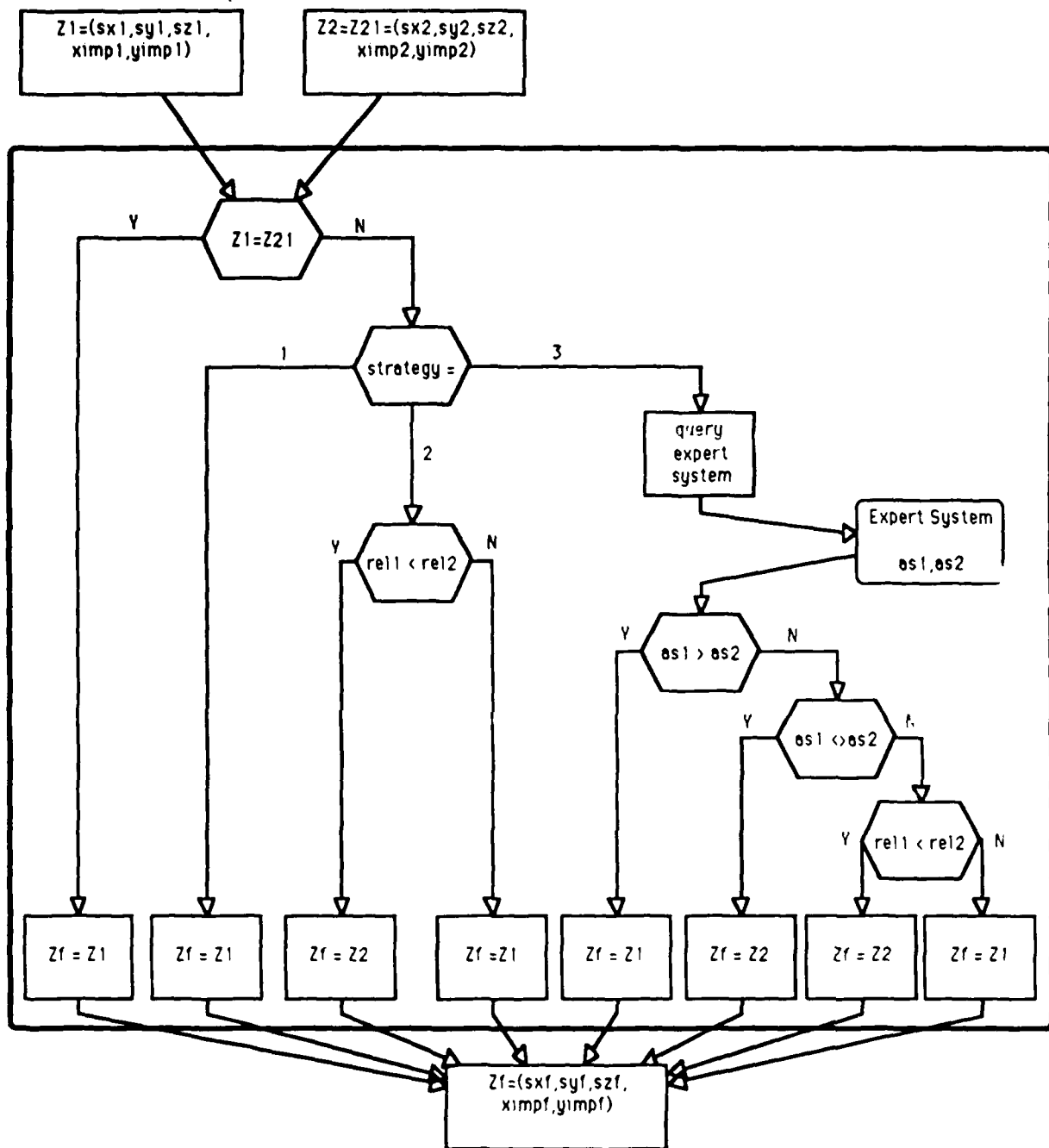
A.4 RESPONSE SELECTION OF DM1

Input : Zf, fused information.

```
         ┌─────────────────┐
         │ Zf=(sxf,syf,szf, │
         │    ximpf,yimpf)  │
         └─────────────────┘
                  │
                  ▽
         ┌─────────────────┐
         │  v=sqrt(sqr(sxf)+│
         │ sqr(syf)+sqr(szf))│
         └─────────────────┘
                  │
                  ▽
         ┌──────────────────┐
         │ d=sqrt(sqr(X-ximpf)+│
         │ sqr(Y-yimpf)+sqr(Z))│
         └──────────────────┘
                  │
                  ▽
         ┌─────────────────┐
         │     t = d/v     │
         └─────────────────┘
                  │
                  ▽
        Y ╱─────────────╲ N
    ┌─────┤   t < tcrit   ├─────┐
    │     ╲─────────────╱     │
    ▽                         ▽
┌──────────────┐      ┌──────────────┐
│  Threat = 1  │      │  Threat = 2  │
└──────────────┘      └──────────────┘
        └──────────┬──────────┘
                   ▽ ▽
         ┌─────────────────────┐
         │ Y1=(threat,sxf,syf,szf)│
         └─────────────────────┘
```
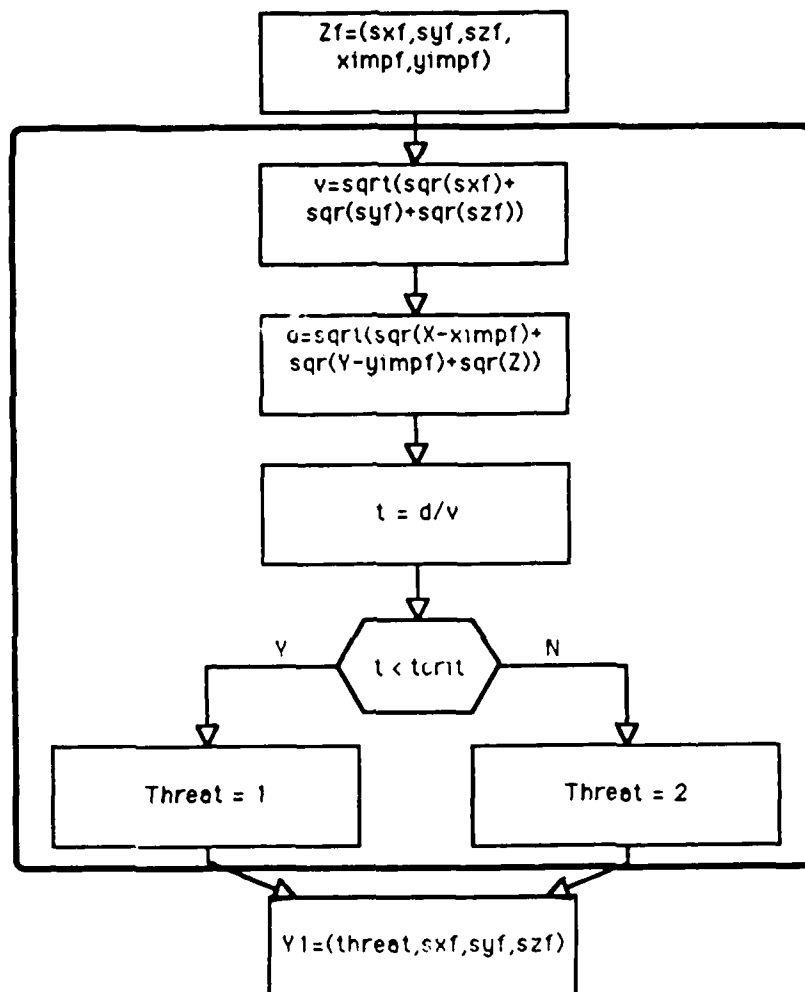
variables

v : speed of the missile

d : distance the missile has to go before reaching the impact point

t : time before reaching the impact point,

tcrit : threshold of response time.

Output :

threat : type of the threat of the missile (1 = very threatening, 2 = less threatening)

sxf,syf,szf : fused velocities

99

A.5 INFORMATION FUSION OF DM2

Input :

    threat : type of the threat of the missile

    sxf,syf,szf : fused velocities

```
        ┌─────────────────────┐
        │   Y1=v=(threat,      │
        │    sxf,syf,szf)      │
        └─────────────────────┘
                  ▽
 ┌───────────────────────────────────────┐
 │           Y  ╱──────────╲  N           │
 │       ┌──────┤ threat = 1 ├──────┐     │
 │       │       ╲──────────╱       │     │
 │       ▽                          ▽     │
 │  ┌──────────┐            ┌──────────┐  │
 │  │ Option = 2 │          │ Option = 1 │ │
 │  └──────────┘            └──────────┘  │
 └───────────────────────────────────────┘
         ╲         ▽      ▽        ╱
        ┌─────────────────────────┐
        │         Option          │
        └─────────────────────────┘
```
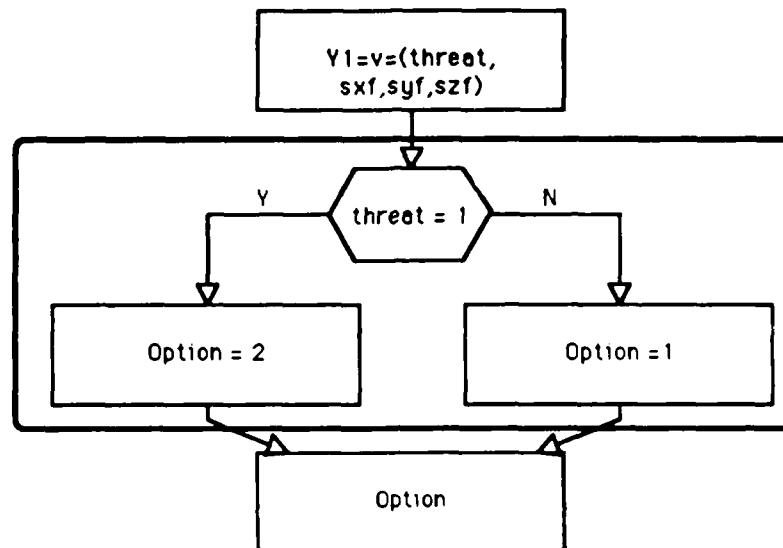
Output :

    Option : Option to follow (1 = use laser beam, 2 = send antimissile rocket).

A.6 RESPONSE SELECTION OF DM2

Input :

Option : Option to follow

sxf, syf, szf : fused velocities

```
                    ┌─────────────────────┐
                    │       Option        │
                    └─────────────────────┘
                              │
        ┌─────────────────────▽─────────────────────────┐
        │                                                │
        │      Y      ╱────────────╲      N              │
        │    ┌────────┤  option = 1 ├────────┐           │
        │    │        ╲────────────╱         │           │
        │    ▽                               ▽           │
        │ ┌──────────────────┐  ┌──────────────────────┐ │
        │ │    weapon =1     │  │     weapon = 2       │ │
        │ │ xtarg = X -(sxf*tresp) │ xtarg = X -(sxf*(tresp+ta)) │
        │ │ ytarg = Y -(syf*tresp) │ ytarg = Y -(syf*(tresp+ta)) │
        │ │ ztarg = Z -(szf*tresp) │ ztarg = Z -(szf*(tresp+ta)) │
        │ └──────────────────┘  └──────────────────────┘ │
        └──────────────┐            ┌────────────────────┘
                       ▽            ▽
              ┌─────────────────────────┐
              │      Y=(weapon,         │
              │    xtarg,ytarg,ztarg)   │
              └─────────────────────────┘
```
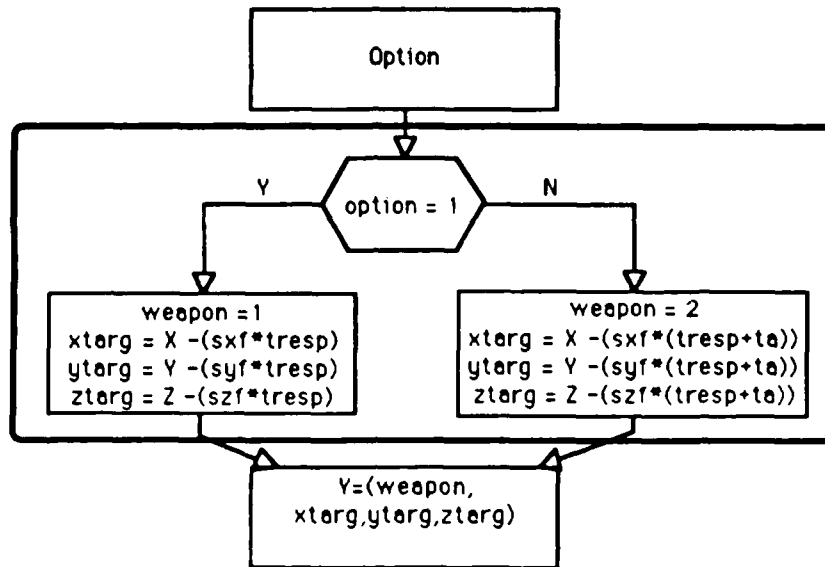
Variables

tresp : estimated response time of the organization

ta : time taken by the antimissile rocket to reach the impact point

Output :

weapon : weapon to use (1 = laser beam, 2 = antimissile rocket)

xtarg, ytarg,ztarg : Coordinates of the position to send the rocket or the laser beam

APPENDIX B

DESCRIPTION OF THE EXPERT SYSTEM
USED IN THE APPLICATION

The expert system used in the application evaluates the severity of the threat of a missile from the coordinates of its impact point entered by the user. It considers, to reach its conclusion, the position of the impact point of the missile relatively to each facility (the three cities, the two production facilities, and the two military bases). For each facility, the system examines first if the missile will reach the facility. If it will, then the value of the threat is 1. If not, the system evaluates the severity of the threat by estimating the distance between the impact point of the missile and the center of the facility. The inference net of this system is shown on the next page.

The degree of truth of each fact is deduced from the input of the decision maker. To check if the missile will reach a specific facility, the system checks just if the coordinates of the impact point are within the perimeter delimiting the facility. A degree of truth of 1 is attributed if it is true, 0 if it is not. To evaluate the degree of truth of the nearness of the target of the missile from a facility, the system computes the distance between the impact point and the center of the facility and executes the mapping function shown on the table below :

Distance in Miles	0-1	2	3	4	5	6	7-8	9	10-11	12-14	>15
Degree of Truth	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0

104
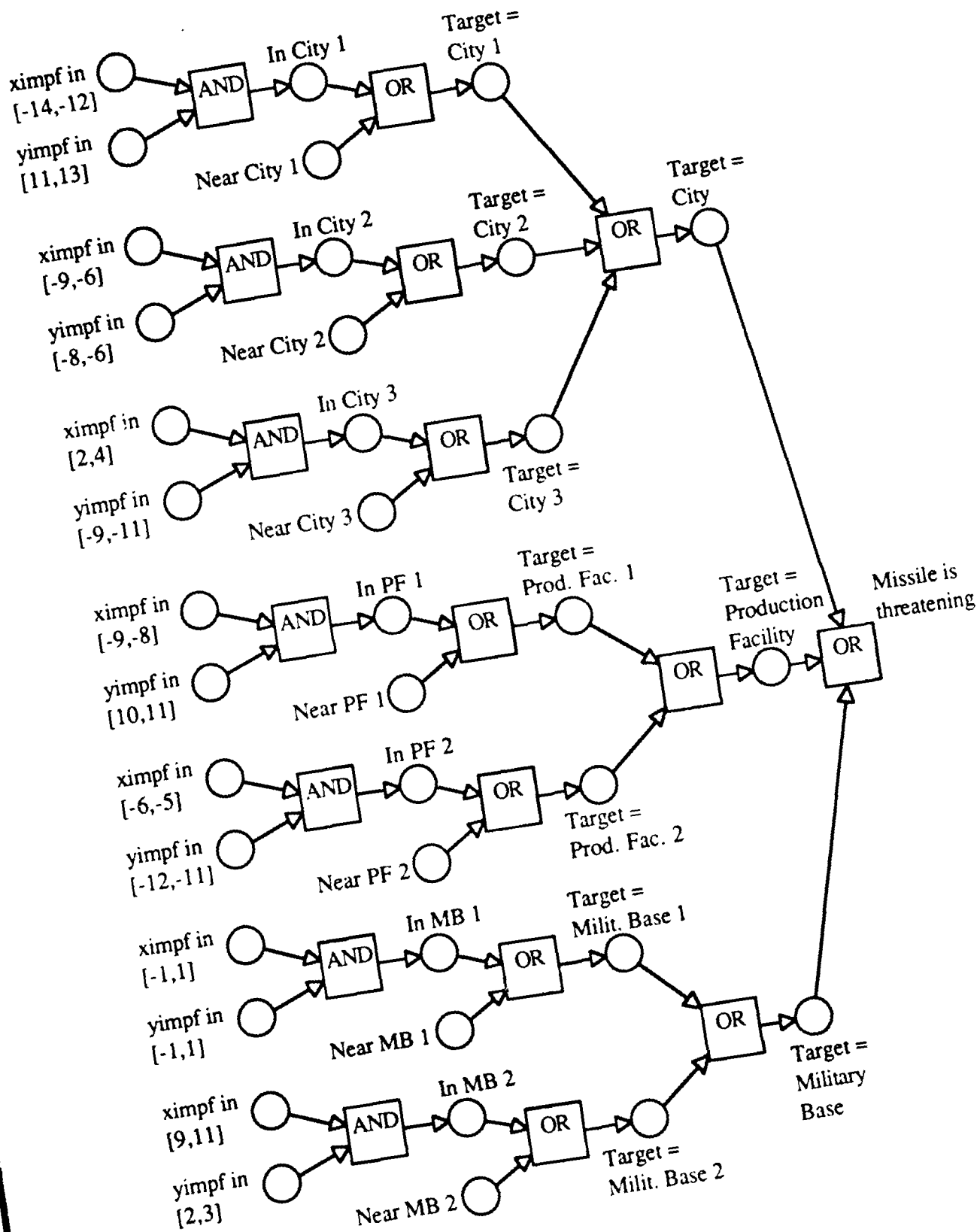
A sequence of operations OR is performed on each of these degrees of truth, which allows the expert system to stop its search as soon as it has assessed that the degree of threat to one of the facilities is larger than 0.8. In the contrary case, all the rules are executed and the degree of threat of the missile is the maximum of the degrees of threat to each facility.

In the example, the system is used by DM1, performing the information fusion, first to assess the degree of threat of the missile according to the trajectory he has determined in his situation assessment stage. The trajectory determined by DM2 is considered by the system only if the answer of the expert system is less than 0.8.

The system with boolean logic uses the same rules. The degree of truth of nearness from each of the facility can take only the value 0 or 1. The value 1 is attributed when the distance from the impact point to the center of the facility is less than 5 miles.

The system looks simple and could have been replaced by a conventional program. Neverthless, it allows to highlight the particularities of such systems which are esssentially the clear distinction between the knowledge base and the control strategy. It shows the pruning of parts of the rule base for checking a specific conclusion and the opening of interaction sessions with the user according to the current state of its search. Moreover, it differs from a conventional program, in that this partition allows for incremental addition of knowledge without manipulating the overall program structure. Such additional knowledge could be, in this case, the geographical aspect of the area, the taking into account of intelligence reports or the interest for the enemy to destroy a specific facility.

END
DATE
FILMED

6- 1988
DTIC